

# PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2001-256066

(43)Date of publication of application : 21.09.2001

(51)Int.Cl.

G06F 9/46

(21)Application number : 2000-054064

(71)Applicant : INTERNATL BUSINESS MACH  
CORP <IBM>

(22)Date of filing : 29.02.2000

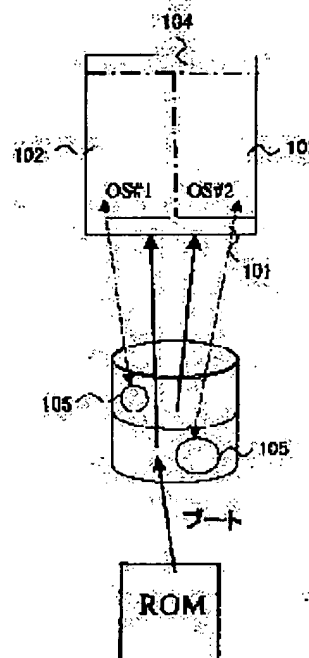
(72)Inventor : SHIMOTOONO SUSUMU

(54) COMPUTER SYSTEM, SWITCHING SYSTEM OF OPERATING SYSTEM, MOUNTING METHOD OF OPERATING SYSTEM, SWITCHING METHOD OF OPERATING SYSTEM, STORAGE MEDIUM AND PROGRAM TRANSMITTER

(57)Abstract:

PROBLEM TO BE SOLVED: To provide environment in which plural operating systems made to coexist in a single system can be used by switching them at high speed.

SOLUTION: In this computer system on which the plural operating systems are mounted, plural memory areas 102, 103 made to individually correspond to each operating system by logically separating a memory area of a memory device and an independent memory area 104 not to be made to correspond to any operating system, an interface means to be operated on the operating system and to issue a switching instruction of the operating systems, a suspend control means, a resume control means and an operating system switching control means to receive the switching instruction from the interface means and to resume the operating system at switching destination after waiting for the operating system to issue the instruction is suspended, are provided.



## LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's  
decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2003 Japan Patent Office

(19)日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開2001-256066

(P2001-256066A)

(43)公開日 平成13年9月21日(2001.9.21)

| (51)Int.Cl. <sup>7</sup> | 識別記号  | F I          | テマコード(参考)       |
|--------------------------|-------|--------------|-----------------|
| G 0 6 F 9/46             | 3 5 0 | G 0 6 F 9/46 | 3 5 0 5 B 0 9 8 |

審査請求 未請求 請求項の数16 O L (全 35 頁)

(21)出願番号 特願2000-54064(P2000-54064)

(22)出願日 平成12年2月29日(2000.2.29)

(71)出願人 390009531

インターナショナル・ビジネス・マシー  
ズ・コーポレーション

INTERNATIONAL BUSIN  
ESS MACHINES CORPO  
RATION

アメリカ合衆国10504、ニューヨーク州  
アーモンク (番地なし)

(74)復代理人 100104880

弁理士 古部 次郎 (外3名)

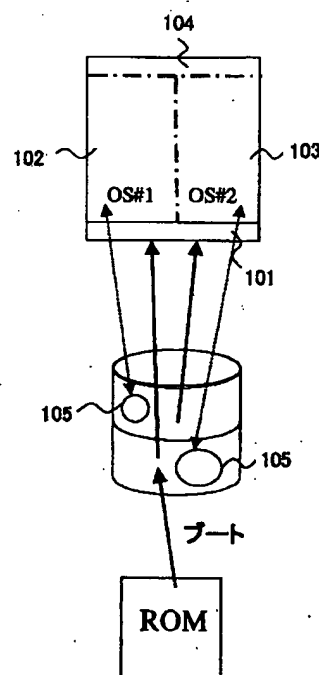
最終頁に続く

(54)【発明の名称】 コンピュータシステム、オペレーティングシステムの切り替えシステム、オペレーティングシステムの実装方法、オペレーティングシステムの切り替え方法、記憶媒体及びプログラム伝送装置

(57)【要約】

【課題】 単一システム内に共存させた複数のオペレーティングシステムを高速に切り替えて使用できる環境を提供する。

【解決手段】 複数のオペレーティングシステムを搭載したコンピュータシステムにおいて、メモリ装置のメモリ領域を論理分割して、各オペレーティングシステムに個別に対応付けられた複数のメモリ領域102、103と、いずれのオペレーティングシステムにも対応付けられない独立したメモリ領域104とを設け、オペレーティングシステム上で動作し、オペレーティングシステムの切り替え指示を発行するインタフェース手段と、サスペンド制御手段と、レジューム制御手段と、インタフェース手段からの切り替え指示を受け取り、この切り替え指示を発行したオペレーティングシステムがサスペンドするのを待って、切り替え先オペレーティングシステムをレジュームさせるオペレーティングシステム切り替え制御手段とを備える。



## 【特許請求の範囲】

【請求項1】 複数のオペレーティングシステムを搭載し、各オペレーティングシステムを切り替えて使用するコンピュータシステムにおいて、メモリ装置のメモリ領域を論理分割して、複数の前記オペレーティングシステムに個別に対応付けられた複数のオペレーティングシステム用メモリ領域と、複数の前記オペレーティングシステムのいずれにも対応付けられない独立したメモリ領域とを設け、前記オペレーティングシステム用メモリ領域に個別にロードされた前記オペレーティングシステム上で動作し、使用するオペレーティングシステムの切り替え指示を発行するためのインタフェース手段と、前記オペレーティングシステム用メモリ領域に個別にロードされた前記オペレーティングシステム上で動作し、前記オペレーティングシステムの動作コンテキストに関する情報を自オペレーティングシステムに割り当てられたオペレーティングシステム用メモリ領域に保存して当該オペレーティングシステムを一時停止状態にするサスペンド制御手段と、前記オペレーティングシステム用メモリ領域に個別にロードされた前記オペレーティングシステム上で動作し、一時停止状態にある前記オペレーティングシステムを動作状態に移行させると共に、前記オペレーティングシステム用メモリ領域に保存されている自オペレーティングシステムの動作コンテキストに関する情報に基づいて自オペレーティングシステムが一時停止状態に移行する直前の動作コンテキストを回復させるレジューム制御手段と、複数の前記オペレーティングシステムのいずれからも独立して動作し、前記インタフェース手段からの前記オペレーティングシステムの切り替え指示を受け取り、当該切り替え指示を発行したオペレーティングシステムが前記サスペンド制御手段により一時停止状態に移行するのを待って、当該切り替え指示にて示された切り替え先オペレーティングシステムのレジューム制御手段に対して当該切り替え先オペレーティングシステムを動作状態に移行させるオペレーティングシステム切り替え制御手段とを備えたことを特徴とするコンピュータシステム。

【請求項2】 前記オペレーティングシステム切り替え制御手段は、

複数の前記オペレーティングシステムが前記メモリ装置のメモリ領域のうちの特定の領域を重複して使用する場合に、前記切り替え先オペレーティングシステムを動作状態に移行させる前に、一時停止状態となった前記オペレーティングシステムが当該特定領域にロードした内容を前記独立したメモリ領域または二次記憶装置に退避させることを特徴とする請求項1に記載のコンピュータシステム。

【請求項3】 複数のオペレーティングシステムを搭載

し、各オペレーティングシステムを切り替えて使用するコンピュータシステムにおいて、メモリ装置のメモリ領域を論理分割して、前記オペレーティングシステムに対応付けられたオペレーティングシステム用メモリ領域と、複数の前記オペレーティングシステムのいずれにも対応付けられない独立したメモリ領域とを設け、

複数の前記オペレーティングシステムのうちで、前記コンピュータシステムの起動時に最初に前記オペレーティングシステム用メモリ領域にブートされるメイン・オペレーティングシステムが、

前記オペレーティングシステム用メモリ領域上に、他の前記オペレーティングシステムをロードするためのメモリ領域を仮想記憶メモリとして確保する仮想記憶メモリ管理手段と、

前記オペレーティングシステムの動作コンテキストに関する情報をオペレーティングシステム用メモリ領域に保存して当該オペレーティングシステムを一時停止状態にするサスペンド制御手段と、

一時停止状態にある前記オペレーティングシステムを動作状態に移行させると共に、オペレーティングシステム用メモリ領域に保存されている自オペレーティングシステムの動作コンテキストに関する情報に基づいて自オペレーティングシステムが一時停止状態に移行する直前の動作コンテキストを回復させるレジューム制御手段とを備え、

前記メイン・オペレーティングシステムから他のオペレーティングシステムに切り替える際に、当該仮想記憶メモリ管理手段にて切り替え先オペレーティングシステム用に仮想記憶メモリによるメモリ領域を確保した後に、前記サスペンド制御手段により当該オペレーティングシステムを一時停止状態にして、当該仮想記憶メモリによるメモリ領域に当該切り替え先オペレーティングシステムをロードし、

前記他のオペレーティングシステムから前記メイン・オペレーティングシステムに切り替える際に、前記他のオペレーティングシステムが停止するのを待って、前記レジューム制御手段により当該メイン・オペレーティングシステムを動作状態に移行させることを特徴とするコンピュータシステム。

【請求項4】 複数のオペレーティングシステムを搭載したコンピュータシステム上で、使用するオペレーティングシステムの切り替えを実行するオペレーティングシステムの切り替えシステムにおいて、

メモリ装置のメモリ領域を論理分割し、複数の前記オペレーティングシステムに個別に対応付けられた複数のオペレーティングシステム用メモリ領域と、複数の前記オペレーティングシステムのいずれにも対応付けられない独立したメモリ領域とを設け、

複数の前記オペレーティングシステム用メモリ領域にそ

れぞれロードされたオペレーティングシステムと、前記独立したメモリ領域にロードされた前記オペレーティングシステムの切り替えを制御するオペレーティングシステム切り替え制御手段とを備え、自オペレーティングシステムの動作コンテキストに関する情報を自オペレーティングシステムに割り当てられたオペレーティングシステム用メモリ領域に保存して当該オペレーティングシステムを一時停止状態にするサスペンド機能と、一時停止状態にある自オペレーティングシステムを動作状態に移行させると共に、オペレーティングシステム用メモリ領域に保存されている自オペレーティングシステムの動作コンテキストに関する情報に基づいて自オペレーティングシステムが一時停止状態に移行する直前の動作コンテキストを回復させるレジューム機能とを有するオペレーティングシステムに関して、前記オペレーティングシステム切り替え制御手段は、使用するオペレーティングシステムを前記オペレーティングシステムから他のオペレーティングシステムに切り替えるために当該オペレーティングシステムを停止させる場合に、前記サスペンド機能を用いて当該オペレーティングシステムを一時停止状態とし、使用するオペレーティングシステムを他のオペレーティングシステムから当該オペレーティングシステムに切り替えるために当該オペレーティングシステムを動作状態に移行させる場合に、前記レジューム機能を用いて当該オペレーティングシステムを動作状態に移行させることを特徴とするオペレーティングシステムの切り替えシステム。

【請求項5】 前記サスペンド機能及び前記レジューム機能を有さないオペレーティングシステムに関して、前記オペレーティングシステム切り替え制御手段は、使用するオペレーティングシステムを前記オペレーティングシステムから他のオペレーティングシステムに切り替えるために当該オペレーティングシステムを停止させる場合に、当該オペレーティングシステムを単独でシャットダウンし、使用するオペレーティングシステムを他のオペレーティングシステムから前記オペレーティングシステムに切り替えるために当該オペレーティングシステムを動作状態に移行させる場合に、当該オペレーティングシステムをリブートすることを特徴とする請求項4に記載のオペレーティングシステムの切り替えシステム。

【請求項6】 複数のオペレーティングシステムを搭載し、各オペレーティングシステムを切り替えて使用するコンピュータシステムにおけるオペレーティングシステムの実装方法において、メモリ装置の所定のメモリ領域に、複数の前記オペレーティングシステムのブート及び切り替えを統括的に制御するプログラムモジュールである統括制御手段をロードするステップと、

前記統括制御手段の制御の下、前記メモリ装置において複数の前記オペレーティングシステムに個別に対応付けられた複数のオペレーティングシステム用メモリ領域に、複数の前記オペレーティングシステムを順次ロードするステップとを含み、

前記オペレーティングシステムを順次ロードするステップは、

所定のオペレーティングシステムのブートローダを用いて当該オペレーティングシステムを当該オペレーティングシステムに対応付けられた前記オペレーティングシステム用メモリ領域にブートするステップと、

さらにロードすべき他のオペレーティングシステムが存在する場合に、ブートした後に前記オペレーティングシステムを停止状態とするステップと、

ブートしたオペレーティングシステムが停止状態となった後に、メモリ装置及びメモリ制御装置を除く他のハードウェアを初期化して前記他のオペレーティングシステムのロード処理へ移行するステップとを含むことを特徴とするオペレーティングシステムの実装方法。

【請求項7】 前記ブート後にオペレーティングシステムを停止状態とするステップは、

当該オペレーティングシステムが、自オペレーティングシステムの動作コンテキストに関する情報を自オペレーティングシステムに割り当てられたオペレーティングシステム用メモリ領域に保存して当該オペレーティングシステムを一時停止状態にするサスペンド機能と、一時停止状態にある自オペレーティングシステムを動作状態に移行させると共に、オペレーティングシステム用メモリ領域に保存されている自オペレーティングシステムの動作コンテキストに関する情報に基づいて自オペレーティングシステムが一時停止状態に移行する直前の動作コンテキストを回復させるレジューム機能とを有する場合に、当該サスペンド機能により当該オペレーティングシステムを一時停止状態に移行させるステップと、

当該オペレーティングシステムが、前記サスペンド機能と前記レジューム機能とを有さない場合に、当該オペレーティングシステムを単独でシャットダウンするステップとを含むことを特徴とする請求項6に記載のオペレーティングシステムの実装方法。

【請求項8】 複数のオペレーティングシステムを搭載し、各オペレーティングシステムを切り替えて使用するコンピュータシステムにおけるオペレーティングシステムの実装方法において、

所定のメモリ領域を仮想記憶メモリとして確保する仮想記憶メモリ管理機能を有するオペレーティングシステムをメモリ装置にブートするステップと、

他のオペレーティングシステムを使用するために、ブートされた前記オペレーティングシステムが、前記仮想記憶メモリ管理機能により所定サイズのメモリ領域を仮想記憶メモリとして確保するステップと、

10

20

30

40

50

仮想記憶メモリとしてメモリ領域を確保した後に、ブートされた前記オペレーティングシステムの動作コンテキストに関する情報を前記メモリ領域に保存して当該オペレーティングシステムを一時停止状態にするステップと、

前記オペレーティングシステムが一時停止状態となった後に、前記仮想記憶メモリとして確保されたメモリ領域に他のオペレーティングシステムをブートするステップと、

前記他のオペレーティングシステムの使用を終了した後、当該他のオペレーティングシステムを停止させ、一時停止状態にある前記オペレーティングシステムを動作状態に移行させると共に、メモリ領域に保存されている当該オペレーティングシステムの動作コンテキストに関する情報に基づいて当該オペレーティングシステムが一時停止状態に移行する直前の動作コンテキストを回復させるステップとを含むことを特徴とするオペレーティングシステムの実装方法。

【請求項9】 複数のオペレーティングシステムを搭載したコンピュータシステム上で、使用するオペレーティングシステムの切り替えを行うオペレーティングシステムの切り替え方法において、

メモリ装置のメモリ領域を論理分割して設けられた複数のオペレーティングシステム用メモリ領域に個別に対応付けられてロードされている複数の前記オペレーティングシステムのうちで、動作状態にあるオペレーティングシステムから他のオペレーティングシステムへの切り替え要求を発行するステップと、

前記切り替え要求を発行したオペレーティングシステムを停止状態に移行させるステップと、

切り替え先のオペレーティングシステムを動作状態に移行させるステップとを含み、

前記オペレーティングシステムを停止状態に移行させるステップは、

前記切り替え要求を発行したオペレーティングシステムが、自オペレーティングシステムの動作コンテキストに関する情報を自オペレーティングシステムに割り当てられたオペレーティングシステム用メモリ領域に保存して当該オペレーティングシステムを一時停止状態にするサスペンド機能を有する場合に、当該サスペンド機能により当該オペレーティングシステムを一時停止状態に移行させるステップを含み、

前記オペレーティングシステムを動作状態に移行させるステップは、

前記切り替え先のオペレーティングシステムが、サスペンド機能により一時停止状態にある自オペレーティングシステムを動作状態に移行させると共に、オペレーティングシステム用メモリ領域に保存されている自オペレーティングシステムの動作コンテキストに関する情報に基づいて自オペレーティングシステムが一時停止状態に移

行する直前の動作コンテキストを回復させるレジューム機能を有する場合に、当該レジューム機能を用いて当該切り替え先のオペレーティングシステムを動作状態に移行させるステップを含むことを特徴とするオペレーティングシステムの切り替え方法。

【請求項10】 複数のオペレーティングシステムを搭載したコンピュータシステム上で、使用するオペレーティングシステムの切り替えを行うオペレーティングシステムの切り替え方法において、

メモリ装置のメモリ領域を論理分割して設けられた複数のオペレーティングシステム用メモリ領域に個別に対応付けられてロードされている複数の前記オペレーティングシステムのうちで、動作状態にあるオペレーティングシステムから他のオペレーティングシステムへの切り替え要求を発行するステップと、

前記切り替え要求を発行したオペレーティングシステムを停止状態に移行させるステップと、

切り替え先のオペレーティングシステムを動作状態に移行させるステップとを含み、

前記オペレーティングシステムを停止状態に移行させるステップは、

前記切り替え要求を発行したオペレーティングシステムが、自オペレーティングシステムの動作コンテキストに関する情報を自オペレーティングシステムに割り当てられたオペレーティングシステム用メモリ領域に保存して当該オペレーティングシステムを一時停止状態にするサスペンド機能を有さない場合に、当該オペレーティングシステムを単独でシャットダウンすることにより停止状態に移行させるステップを含み、

前記オペレーティングシステムを動作状態に移行させるステップは、

前記切り替え先のオペレーティングシステムが、サスペンド機能により一時停止状態にある自オペレーティングシステムを動作状態に移行させると共に、オペレーティングシステム用メモリ領域に保存されている自オペレーティングシステムの動作コンテキストに関する情報に基づいて自オペレーティングシステムが一時停止状態に移行する直前の動作コンテキストを回復させるレジューム機能を有さない場合に、当該切り替え先のオペレーティングシステムをリブートすることにより動作状態に移行させるステップを含むことを特徴とするオペレーティングシステムの切り替え方法。

【請求項11】 コンピュータに実行させるプログラムを当該コンピュータの入力手段が読取可能に記憶した記憶媒体において、

前記プログラムは、

メモリ装置のメモリ領域を論理分割し、複数の前記オペレーティングシステムに個別に対応付けられた複数のオペレーティングシステム用メモリ領域を設ける処理と、複数の前記オペレーティングシステム用メモリ領域に、

対応付けられた前記オペレーティングシステムを順次ロードする処理として、所定のオペレーティングシステムのブートローダを用いて当該オペレーティングシステムを当該オペレーティングシステムに対応付けられた当該オペレーティングシステム用メモリ領域にブートする処理と、さらにロードすべき他のオペレーティングシステムが存在する場合に、ブートした後に当該オペレーティングシステムを停止状態とする処理と、ブートしたオペレーティングシステムが停止状態となった後に、メモリ装置及びメモリ制御装置を除く他のハードウェアを初期化して前記他のオペレーティングシステムのロード処理へ移行する処理とを前記コンピュータに実行させることを特徴とする記憶媒体。

【請求項 1 2】 前記プログラムにおける前記ブート後にオペレーティングシステムを停止状態とする処理は、当該オペレーティングシステムが、自オペレーティングシステムの動作コンテキストに関する情報を自オペレーティングシステムに割り当てられたオペレーティングシステム用メモリ領域に保存して当該オペレーティングシステムを一時停止状態にするサスペンド機能と、一時停止状態にある自オペレーティングシステムを動作状態に移行させると共に、オペレーティングシステム用メモリ領域に保存されている自オペレーティングシステムの動作コンテキストに関する情報に基づいて自オペレーティングシステムが一時停止状態に移行する直前の動作コンテキストを回復させるレジューム機能とを有する場合に、当該サスペンド機能により当該オペレーティングシステムを一時停止状態に移行させる処理と、当該オペレーティングシステムが、前記サスペンド機能と前記レジューム機能とを有さない場合に、当該オペレーティングシステムを単独でシャットダウンする処理とを含むことを特徴とする請求項 1 1 に記載の記憶媒体。

【請求項 1 3】 コンピュータに実行させるプログラムを当該コンピュータの入力手段が読取可能に記憶した記憶媒体において、前記プログラムは、メモリ装置のメモリ領域を論理分割して設けられた複数のオペレーティングシステム用メモリ領域に個別に対応付けられてロードされている複数の前記オペレーティングシステムのうちで、動作状態にあるオペレーティングシステムから他のオペレーティングシステムへの切り替え要求を受け付ける処理と、当該切り替え要求を発行したオペレーティングシステムが、自オペレーティングシステムの動作コンテキストに関する情報を自オペレーティングシステムに割り当てられたオペレーティングシステム用メモリ領域に保存して当該オペレーティングシステムを一時停止状態にするサスペンド機能を有する場合に、当該サスペンド機能により当該オペレーティングシステムを一時停止状態に移行させる処理と、

切り替え先のオペレーティングシステムが、サスペンド機能により一時停止状態にある自オペレーティングシステムを動作状態に移行させると共に、オペレーティングシステム用メモリ領域に保存されている自オペレーティングシステムの動作コンテキストに関する情報に基づいて自オペレーティングシステムが一時停止状態に移行する直前の動作コンテキストを回復させるレジューム機能を有する場合に、当該レジューム機能を用いて当該切り替え先のオペレーティングシステムを動作状態に移行させる処理とを前記コンピュータに実行させることを特徴とする記憶媒体。

【請求項 1 4】 コンピュータに、メモリ装置のメモリ領域を論理分割し、複数の前記オペレーティングシステムに個別に対応付けられた複数のオペレーティングシステム用メモリ領域を設ける処理と、複数の前記オペレーティングシステム用メモリ領域に、対応付けられた前記オペレーティングシステムを順次ロードする処理として、所定のオペレーティングシステムのブートローダを用いて当該オペレーティングシステムを当該オペレーティングシステムに対応付けられた当該オペレーティングシステム用メモリ領域にブートする処理と、さらにロードすべき他のオペレーティングシステムが存在する場合に、ブートした後に当該オペレーティングシステムを停止状態とする処理と、ブートしたオペレーティングシステムが停止状態となった後に、メモリ装置及びメモリ制御装置を除く他のハードウェアを初期化して前記他のオペレーティングシステムのロード処理へ移行する処理とを実行させるプログラムを記憶する記憶手段と、

前記記憶手段から前記プログラムを読み出して当該プログラムを送信する送信手段とを備えたことを特徴とするプログラム伝送装置。

【請求項 1 5】 前記プログラムにおける前記ブート後にオペレーティングシステムを停止状態とする処理は、当該オペレーティングシステムが、自オペレーティングシステムの動作コンテキストに関する情報を自オペレーティングシステムに割り当てられたオペレーティングシステム用メモリ領域に保存して当該オペレーティングシステムを一時停止状態にするサスペンド機能と、一時停止状態にある自オペレーティングシステムを動作状態に移行させると共に、オペレーティングシステム用メモリ領域に保存されている自オペレーティングシステムの動作コンテキストに関する情報に基づいて自オペレーティングシステムが一時停止状態に移行する直前の動作コンテキストを回復させるレジューム機能とを有する場合に、当該サスペンド機能により当該オペレーティングシステムを一時停止状態に移行させる処理と、当該オペレーティングシステムが、前記サスペンド機能と前記レジューム機能とを有さない場合に、当該オペレーティングシステムを単独でシャットダウンする処理と

を含むことを特徴とする請求項14に記載のプログラム伝送装置。

【請求項16】 コンピュータに、  
メモリ装置のメモリ領域を論理分割して設けられた複数のオペレーティングシステム用メモリ領域に個別に対応付けられてロードされている複数の前記オペレーティングシステムのうちで、動作状態にあるオペレーティングシステムから他のオペレーティングシステムへの切り替え要求を受け付ける処理と、当該切り替え要求を発行したオペレーティングシステムが、自オペレーティングシステムの動作コンテキストに関する情報を自オペレーティングシステムに割り当てられたオペレーティングシステム用メモリ領域に保存して当該オペレーティングシステムを一時停止状態にするサスペンド機能を有する場合に、当該サスペンド機能により当該オペレーティングシステムを一時停止状態に移行させる処理と、切り替え先のオペレーティングシステムが、サスペンド機能により一時停止状態にある自オペレーティングシステムを動作状態に移行させると共に、オペレーティングシステム用メモリ領域に保存されている自オペレーティングシステムの動作コンテキストに関する情報に基づいて自オペレーティングシステムが一時停止状態に移行する直前の動作コンテキストを回復させるレジューム機能を有する場合に、当該レジューム機能を用いて当該切り替え先のオペレーティングシステムを動作状態に移行させる処理とを実行させるプログラムを記憶する記憶手段と、前記記憶手段から前記プログラムを読み出して当該プログラムを送信する送信手段とを備えたことを特徴とするプログラム伝送装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 本発明は、単一のコンピュータシステム内に複数のオペレーティングシステムを共存させ、切り替えながら使用するシステム制御技術に関する。

【0002】

【従来の技術】 近年、コンピュータの利用環境の増加に伴い、一つのコンピュータシステム内に複数のオペレーティングシステム（以下、OSと略す）を共存させ、切り替えながら使用する使用態様が現れてきている。従来、このような複数のOSを共存させる環境では、OSの切り替え方式として、一般に、ブート時にOSを選択的に切り替える方式や、エミュレーションにより所定のOSに他のOSの機能を持たせる方式があった。

【0003】 ブート時にOSを切り替える方式では、コンピュータをブートする際に、起動可能な複数のOSの中からユーザが手動で所望のOSを選択し起動させる。この種の方式をサポートしたツールは、通常ブートセ

レクタの方式としては、大別して以下の三つがある。

(1) ROMコードがOSへ制御を移す際には、ハードディスクの先頭セクタ (MBR : Master Boot Record) に機械的にアクセスして所定のアドレスのメモリ領域に同セクタの内容をロードし、さらにそのロードイメージの先頭にジャンプする。そこで、この一連の動作に介入してOSの切り替えを行う方式。すなわち、MBRの内容を予め操作 (入れ替え) して、OSを起動する動作の最初の段階に介入し、複数のOSの起動を可能とする。

(2) 所定のOSのブートローダとして一旦起動されたのち、当該ブートローダの機能において、他のOSを起動する方式。

(3) 起動パーティションとして独自の基本区画を占有し、ROMコードから見ると一つのOSとして立ち上げられ、その中で他のOSを起動する方式。

【0004】 この種のOSの切り替え方式では、いずれの場合も、複数のOSの個々の本体イメージは、ハードディスク内のそれぞれ独立した区画 (Partition) に保存されている。そして、ブートセクタの方式としては、特定の区画に保存されたOSのみを起動可能なものや、単一のハードディスクにおける何れかの区画に保存されたOSを選択して起動可能なもの、複数のハードディスクにおける何れかの区画に保存されたOSを選択して起動可能なものなどがある。

【0005】 エミュレーションにより所定のOSに他のOSの機能を持たせる方式では、所定のOSの動作環境に他のOSの動作環境の一部を仮想的に構築し、前提とするOS環境が異なるアプリケーションであっても単一のOS環境で動作できるようにする。すなわち、この方式は、複数のOSそのものを共存させるものではなく、各OSのサービス (API : Application Program Interface) に依存したアプリケーションが別のOSで動作可能なように、当該サービス提供レイヤと同等の機能を持つレイヤを実現したに過ぎない。

【0006】

【発明が解決しようとする課題】 しかしながら、上述した単一システム上で複数のOSを切り替えて使用する従来の技術は、以下のような欠点を有する。

【0007】 ブート時にOSを切り替える方式では、まず、単一システム内での複数OSの共存は実現されているものの、その切り替えは、互いに再起動 (リブート) することによって行うため、非常に時間がかかるということである。すなわち、OSを切り替える際には、現行のOSをシャットダウンし、ROMを起動し、続いて次の起動OSを起動するまでの一連の処理が行われ、各処理に要する時間を経てようやく完了する。このため、実用上の使い勝手が著しく悪かった。

【0008】 また、OSを切り替えると、それまで使用していたOSをシャットダウンするため、当該OSは起動した直後 (デフォルト) の状態に戻ってしまう。その



ため、再びOSを切り替えて当該OSへ戻ってきたとしても、切り替える直前の状態（コンテキスト）は保存されておらず、失われてしまう。これは、頻繁にOSの切り替えを行って複数のOSを使い分けるユーザにとっては極めて使い勝手が悪かった。例えば、OSを切り替える直前まで利用していたアプリケーションは、OSの切り替えに際して、その処理中のデータファイルを一旦終了して保存する必要がある。そして、当該OSに復帰した際に、当該アプリケーションを再度立ち上げ、当該データファイルを読み込み、注目していたページの位置まで手動で元に戻す作業が必要になる。これでは、一連の作業の最中にOSの切り替えを行うことは、その作業の中断に等しい。したがって、とても各OSに特有のアプリケーションを複数利用してデータファイル間で連携するような使い方ができる環境ではない。

【0009】一方、エミュレーションにより所定のOSに他のOSの機能を持たせる方式では、OSそのものをシステム中に共存させるのではなく、複数OSの動作環境をエミュレーションにより仮想的に構築したに過ぎないため、互換性が完全ではなかった。また、OSのバージョンアップなどに逐次対応して、アプリケーションレイヤ向けに実現された仮想OS環境の互換性を維持、検証する作業が必要であった。さらにまた、完全に別個のOSの動作環境を共存させているわけではないため、OS固有のシステムサービスを利用したり、同じくOS固有の資源を前提にするアプリケーションサーバーやカーネル空間で動作するデバイスドライバ及びデバイスを当該仮想環境で動作させたりすることができなかった。

【0010】そこで本発明は、単一システム内に共存させた複数のオペレーティングシステムを高速に切り替えて使用できる環境を提供することを目的とする。

【0011】また本発明は、単一システム内に共存させた複数のオペレーティングシステムを切り替えた際に、切り替え前の状態を保存しておき、再度当該オペレーティングシステムに切り替えた際に、切り替え前の状態を復帰できるようにすることを他の目的とする。

【0012】

【課題を解決するための手段】かかる目的のもと、本発明は、複数のオペレーティングシステムを搭載し、各オペレーティングシステムを切り替えて使用するコンピュータシステムにおいて、メモリ装置のメモリ領域を論理分割して、複数の前記オペレーティングシステムに個別に対応付けられた複数のオペレーティングシステム用メモリ領域と、複数のオペレーティングシステムのいずれにも対応付けられない独立したメモリ領域とを設け、オペレーティングシステム用メモリ領域に個別にロードされたオペレーティングシステム上で動作し、使用するオペレーティングシステムの切り替え指示を発行するためのインタフェース手段と、オペレーティングシステム用メモリ領域に個別にロードされたオペレーティングシ

テム上で動作し、オペレーティングシステムの動作コンテキストに関する情報を自オペレーティングシステムに割り当てられたオペレーティングシステム用メモリ領域に保存してこのオペレーティングシステムを一時停止状態にするサスペンド制御手段と、オペレーティングシステム用メモリ領域に個別にロードされたオペレーティングシステム上で動作し、一時停止状態にあるオペレーティングシステムを動作状態に移行させると共に、オペレーティングシステム用メモリ領域に保存されている自オペレーティングシステムの動作コンテキストに関する情報に基づいて自オペレーティングシステムが一時停止状態に移行する直前の動作コンテキストを回復させるレジューム制御手段と、複数のオペレーティングシステムのいずれからも独立して動作し、インタフェース手段からのオペレーティングシステムの切り替え指示を受け取り、この切り替え指示を発行したオペレーティングシステムがサスペンド制御手段により一時停止状態に移行するのを待って、この切り替え指示にて示された切り替え先オペレーティングシステムのレジューム制御手段に対してこの切り替え先オペレーティングシステムを動作状態に移行させるオペレーティングシステム切り替え制御手段とを備えたことを特徴としている。

【0013】ここで、オペレーティングシステム切り替え制御手段は、複数のオペレーティングシステムがメモリ装置のメモリ領域のうちの特定の領域を重複して使用する場合に、切り替え先オペレーティングシステムを動作状態に移行させる前に、一時停止状態となったオペレーティングシステムがこの特定領域にロードした内容を独立したメモリ領域または二次記憶装置に退避させることを特徴としている。すなわち、メモリ装置の特定のメモリ領域を複数のオペレーティングシステムが必ず固定的に利用しなければならない場合に、結果としてこれら複数のオペレーティングシステム間で同領域を互いに重複して使用することとなる。このため、オペレーティングシステムを切り替える際には、それまで使用していたオペレーティングシステムにおける当該領域の内容を他の場所に退避させ、次に使用するオペレーティングシステムのために空き領域としなければならない。その退避先として、メモリ装置における独立したメモリ領域または二次記憶装置を使用することができる。このような構成とすれば、特定の領域を重複して使用するオペレーティングシステムが存在する場合にも、メモリ装置上に共存させることができる点で優れている。なお、メモリ装置または二次記憶装置に退避された内容は、一時停止状態にあるオペレーティングシステムがレジューム制御手段の制御により再稼働する前に元のメモリ領域（すなわち重複して使用される領域）に回復させる。また、二次記憶装置を介して退避・回復する方法は、複数のOSに重複利用されるメモリ領域が大きい場合には、それに合わせて独立したメモリ領域を大きく確保する必要がない

という点で優れている。一方、独立したメモリ領域のみで退避・回復する方法は処理速度が速いという点で優れている。

【0014】また、本発明は、複数のオペレーティングシステムを搭載し、各オペレーティングシステムを切り替えて使用するコンピュータシステムにおいて、メモリ装置のメモリ領域を論理分割して、オペレーティングシステムに対応付けられたオペレーティングシステム用メモリ領域と、複数のオペレーティングシステムのいずれにも対応付けられない独立したメモリ領域とを設け、複数のオペレーティングシステムのうちで、コンピュータシステムの起動時に最初にオペレーティングシステム用メモリ領域にブートされるメイン・オペレーティングシステムが、オペレーティングシステム用メモリ領域上に、他のオペレーティングシステムをロードするためのメモリ領域を仮想記憶メモリとして確保する仮想記憶メモリ管理手段と、オペレーティングシステムの動作コンテキストに関する情報をオペレーティングシステム用メモリ領域に保存してこのオペレーティングシステムを一時停止状態にするサスペンド制御手段と、一時停止状態にあるオペレーティングシステムを動作状態に移行させると共に、オペレーティングシステム用メモリ領域に保存されている自オペレーティングシステムの動作コンテキストに関する情報に基づいて自オペレーティングシステムが一時停止状態に移行する直前の動作コンテキストを回復させるレジューム制御手段とを備え、メイン・オペレーティングシステムから他のオペレーティングシステムに切り替える際に、この仮想記憶メモリ管理手段にて切り替え先オペレーティングシステム用に仮想記憶メモリによるメモリ領域を確保した後に、サスペンド制御手段によりこのオペレーティングシステムを一時停止状態にして、この仮想記憶メモリによるメモリ領域にこの切り替え先オペレーティングシステムをロードし、他のオペレーティングシステムからメイン・オペレーティングシステムに切り替える際に、当該他のオペレーティングシステムが停止するのを待って、レジューム制御手段によりメイン・オペレーティングシステムを動作状態に移行させることを特徴としている。メイン・オペレーティングシステムは、他のオペレーティングシステムをロードするメモリ領域を保持するために、サスペンド機能及びレジューム機能を用いて他のオペレーティングシステムに切り替わる必要があるが、他のオペレーティングシステムにおいては、サスペンド機能及びレジューム機能を用いても良いし、シャットダウン及びリブートを用いて、さらに他のオペレーティングシステムやメイン・オペレーティングシステムに切り替わるようにしても良い。

【0015】さらにまた、本発明は、複数のオペレーティングシステムを搭載したコンピュータシステム上で、使用するオペレーティングシステムの切り替えを実行す

るオペレーティングシステムの切り替えシステムにおいて、メモリ装置のメモリ領域を論理分割し、複数の前記オペレーティングシステムに個別に対応付けられた複数のオペレーティングシステム用メモリ領域と、複数の前記オペレーティングシステムのいずれにも対応付けられない独立したメモリ領域とを設け、複数の前記オペレーティングシステム用メモリ領域にそれぞれロードされたオペレーティングシステムと、独立したメモリ領域にロードされたオペレーティングシステムの切り替えを制御するオペレーティングシステム切り替え制御手段とを備え、自オペレーティングシステムの動作コンテキストに関する情報を自オペレーティングシステムに割り当てられたオペレーティングシステム用メモリ領域に保存してこのオペレーティングシステムを一時停止状態にするサスペンド機能と、一時停止状態にある自オペレーティングシステムを動作状態に移行させると共に、オペレーティングシステム用メモリ領域に保存されている自オペレーティングシステムの動作コンテキストに関する情報に基づいて自オペレーティングシステムが一時停止状態に移行する直前の動作コンテキストを回復させるレジューム機能とを有するオペレーティングシステムに関して、オペレーティングシステム切り替え制御手段は、使用するオペレーティングシステムをこのオペレーティングシステムから他のオペレーティングシステムに切り替えるためにこのオペレーティングシステムを停止させる場合に、サスペンド機能を用いてこのオペレーティングシステムを一時停止状態とし、使用するオペレーティングシステムを他のオペレーティングシステムからこのオペレーティングシステムに切り替えるためにこのオペレーティングシステムを動作状態に移行させる場合に、レジューム機能を用いてこのオペレーティングシステムを動作状態に移行させることを特徴としている。

【0016】ここで、サスペンド機能及びレジューム機能を有さないオペレーティングシステムに関して、オペレーティングシステム切り替え制御手段は、使用するオペレーティングシステムをこのオペレーティングシステムから他のオペレーティングシステムに切り替えるためにこのオペレーティングシステムを停止させる場合に、このオペレーティングシステムを単独でシャットダウンし、使用するオペレーティングシステムを他のオペレーティングシステムからこのオペレーティングシステムに切り替えるためにこのオペレーティングシステムを動作状態に移行させる場合に、このオペレーティングシステムをリブートすることを特徴としている。オペレーティングシステムの切り替えにシャットダウン及びリブートを用いることにより、動作コンテキストを保存することはできなくなる。また、サスペンド機能及びレジューム機能を用いる場合に比べてオペレーティングシステムを切り替えるために要する時間が大きくなる。一方、シャットダウン及びリブートを用いる利点は、必ずしもすべ

てのオペレーティングシステムが本発明の実装時点でサスペンド・レジューム機能をサポートしているとは限らないため、そのようなオペレーティングシステムにも本発明を適用できることである。特に、動作コンテキストを保存する必要がない場合や、シャットダウン及びリブートが短時間で可能なOSにおいては、シャットダウン及びリブートによるオペレーティングシステムの切り替えでも充分に有効な方法となる。

【0017】また、本発明は、複数のオペレーティングシステムを搭載し、各オペレーティングシステムを切り替えて使用するコンピュータシステムにおけるオペレーティングシステムの実装方法において、メモリ装置の所定のメモリ領域に、複数のオペレーティングシステムのブート及び切り替えを統括的に制御するプログラムモジュールである統括制御手段をロードするステップと、この統括制御手段の制御の下、メモリ装置において複数のオペレーティングシステムに個別に対応付けられた複数のオペレーティングシステム用メモリ領域に、複数のオペレーティングシステムを順次ロードするステップとを含み、オペレーティングシステムを順次ロードするステップは、所定のオペレーティングシステムのブートロードを用いてこのオペレーティングシステムをこのオペレーティングシステムに対応付けられたオペレーティングシステム用メモリ領域にブートするステップと、さらにロードすべき他のオペレーティングシステムが存在する場合に、ブートした後にオペレーティングシステムを停止状態とするステップと、ブートしたオペレーティングシステムが停止状態となった後に、メモリ装置及びメモリ制御装置を除く他のハードウェアを初期化して他のオペレーティングシステムのロード処理へ移行するステップとを含むことを特徴としている。なお、最初にロードされるオペレーティングシステムにおける論理分割メモリの適正な確保は、当該オペレーティングシステムよりも先にロードされる統括制御手段が行う。また、最後にロードするオペレーティングシステムは、停止状態に移行する必要がないので、ブートされた状態のままとなる。すなわち、本発明においてコンピュータシステムを起動すると、最後にロードするオペレーティングシステムが起動完了した状態で通常動作が開始されることとなる。

【0018】ここで、ブート後にオペレーティングシステムを停止状態とするステップは、このオペレーティングシステムが、自オペレーティングシステムの動作コンテキストに関する情報を自オペレーティングシステムに割り当てられたオペレーティングシステム用メモリ領域に保存してこのオペレーティングシステムを一時停止状態にするサスペンド機能と、一時停止状態にある自オペレーティングシステムを動作状態に移行させると共に、オペレーティングシステム用メモリ領域に保存されている自オペレーティングシステムの動作コンテキストに関

する情報に基づいて自オペレーティングシステムが一時停止状態に移行する直前の動作コンテキストを回復させるレジューム機能とを有する場合に、このサスペンド機能によりこのオペレーティングシステムを一時停止状態に移行させるステップと、このオペレーティングシステムが、サスペンド機能とレジューム機能とを有さない場合に、このオペレーティングシステムを単独でシャットダウンするステップとを含むことを特徴としている。

【0019】また、本発明は、複数のオペレーティングシステムを搭載し、各オペレーティングシステムを切り替えて使用するコンピュータシステムにおけるオペレーティングシステムの実装方法において、所定のメモリ領域を仮想記憶メモリとして確保する仮想記憶メモリ管理機能を有するオペレーティングシステムをメモリ装置にブートするステップと、他のオペレーティングシステムを使用するために、ブートされたオペレーティングシステムが、仮想記憶メモリ管理機能により所定サイズのメモリ領域を仮想記憶メモリとして確保するステップと、仮想記憶メモリとしてメモリ領域を確保した後に、ブートされたオペレーティングシステムの動作コンテキストに関する情報をメモリ領域に保存してこのオペレーティングシステムを一時停止状態にするステップと、このオペレーティングシステムが一時停止状態となった後に、仮想記憶メモリとして確保されたメモリ領域に他のオペレーティングシステムをブートするステップと、他のオペレーティングシステムの使用を終了した後、他のオペレーティングシステムを停止させ、一時停止状態にあるオペレーティングシステムを動作状態に移行させると共に、メモリ領域に保存されているこのオペレーティングシステムの動作コンテキストに関する情報に基づいてこのオペレーティングシステムが一時停止状態に移行する直前の動作コンテキストを回復させるステップとを含むことを特徴としている。

【0020】さらにまた、本発明は、複数のオペレーティングシステムを搭載したコンピュータシステム上で、使用するオペレーティングシステムの切り替えを行うオペレーティングシステムの切り替え方法において、メモリ装置のメモリ領域を論理分割して設けられた複数のオペレーティングシステム用メモリ領域に個別に対応付けられてロードされている複数のオペレーティングシステムのうちで、動作状態にあるオペレーティングシステムから他のオペレーティングシステムへの切り替え要求を発行するステップと、この切り替え要求を発行したオペレーティングシステムを停止状態に移行させるステップと、切り替え先のオペレーティングシステムを動作状態に移行させるステップとを含み、オペレーティングシステムを停止状態に移行させるステップは、切り替え要求を発行したオペレーティングシステムが、自オペレーティングシステムの動作コンテキストに関する情報を自オペレーティングシステムに割り当てられたオペレーティ

ングシステム用メモリ領域に保存してこのオペレーティングシステムを一時停止状態にするサスペンド機能を有する場合に、このサスペンド機能によりこのオペレーティングシステムを一時停止状態に移行させるステップを含み、このオペレーティングシステムを動作状態に移行させるステップは、この切り替え先のオペレーティングシステムが、サスペンド機能により一時停止状態にある自オペレーティングシステムを動作状態に移行させると共に、オペレーティングシステム用メモリ領域に保存されている自オペレーティングシステムの動作コンテキストに関する情報に基づいて自オペレーティングシステムが一時停止状態に移行する直前の動作コンテキストを回復させるレジューム機能を有する場合に、このレジューム機能を用いてこの切り替え先のオペレーティングシステムを動作状態に移行させるステップを含むことを特徴としている。

【0021】また、本発明は、複数のオペレーティングシステムを搭載したコンピュータシステム上で、使用するオペレーティングシステムの切り替えを行うオペレーティングシステムの切り替え方法において、メモリ装置のメモリ領域を論理分割して設けられた複数のオペレーティングシステム用メモリ領域に個別に対応付けられてロードされている複数のオペレーティングシステムのうちで、動作状態にあるオペレーティングシステムから他のオペレーティングシステムへの切り替え要求を発行するステップと、切り替え要求を発行したオペレーティングシステムを停止状態に移行させるステップと、切り替え先のオペレーティングシステムを動作状態に移行させるステップとを含み、オペレーティングシステムを停止状態に移行させるステップは、切り替え要求を発行したオペレーティングシステムが、自オペレーティングシステムの動作コンテキストに関する情報を自オペレーティングシステムに割り当てられたオペレーティングシステム用メモリ領域に保存してこのオペレーティングシステムを一時停止状態にするサスペンド機能を有さない場合に、このオペレーティングシステムを単独でシャットダウンすることにより停止状態に移行させるステップを含み、このオペレーティングシステムを動作状態に移行させるステップは、この切り替え先のオペレーティングシステムが、サスペンド機能により一時停止状態にある自オペレーティングシステムを動作状態に移行させると共に、オペレーティングシステム用メモリ領域に保存されている自オペレーティングシステムの動作コンテキストに関する情報に基づいて自オペレーティングシステムが一時停止状態に移行する直前の動作コンテキストを回復させるレジューム機能を有さない場合に、この切り替え先のオペレーティングシステムをリブートすることにより動作状態に移行させるステップを含むことを特徴としている。このような構成とすれば、サスペンド機能及びレジューム機能を有さないオペレーティングシステムで

あっても、他のOSと共存させることができる。

【0022】また、本発明は、コンピュータに実行させるプログラムを当該コンピュータの入力手段が読取可能に記憶した記憶媒体において、このプログラムは、メモリ装置のメモリ領域を論理分割し、複数のオペレーティングシステムに個別に対応付けられた複数のオペレーティングシステム用メモリ領域を設ける処理と、複数のオペレーティングシステム用メモリ領域に、対応付けられたオペレーティングシステムを順次ロードする処理として、所定のオペレーティングシステムのブートロードを用いてこのオペレーティングシステムをこのオペレーティングシステムに対応付けられたオペレーティングシステム用メモリ領域にブートする処理と、さらにロードすべき他のオペレーティングシステムが存在する場合に、ブートした後にこのオペレーティングシステムを停止状態とする処理と、ブートしたオペレーティングシステムが停止状態となった後に、メモリ装置及びメモリ制御装置を除く他のハードウェアを初期化して他のオペレーティングシステムのロード処理へ移行する処理とをこのコンピュータに実行させることを特徴としている。このような構成とすれば、このプログラムをインストールした全てのコンピュータにおいて、オペレーティングシステムを共存させることが可能となる。

【0023】ここで、このプログラムにおけるブート後にオペレーティングシステムを停止状態とする処理は、このオペレーティングシステムが、自オペレーティングシステムの動作コンテキストに関する情報を自オペレーティングシステムに割り当てられたオペレーティングシステム用メモリ領域に保存して当該オペレーティングシステムを一時停止状態にするサスペンド機能と、一時停止状態にある自オペレーティングシステムを動作状態に移行させると共に、オペレーティングシステム用メモリ領域に保存されている自オペレーティングシステムの動作コンテキストに関する情報に基づいて自オペレーティングシステムが一時停止状態に移行する直前の動作コンテキストを回復させるレジューム機能とを有する場合に、このサスペンド機能により当該オペレーティングシステムを一時停止状態に移行させる処理と、このオペレーティングシステムが、サスペンド機能とレジューム機能とを有さない場合に、このオペレーティングシステムを単独でシャットダウンする処理とを含むことを特徴としている。このような構成とすれば、サスペンド機能及びレジューム機能を有さないオペレーティングシステムであっても、他のOSと共存させることができる。

【0024】さらにまた、本発明は、コンピュータに実行させるプログラムを当該コンピュータの入力手段が読取可能に記憶した記憶媒体において、このプログラムは、メモリ装置のメモリ領域を論理分割して設けられた複数のオペレーティングシステム用メモリ領域に個別に対応付けられてロードされている複数のオペレーティン

グシステムのうちで、動作状態にあるオペレーティングシステムから他のオペレーティングシステムへの切り替え要求を受け付ける処理と、この切り替え要求を発行したオペレーティングシステムが、自オペレーティングシステムの動作コンテキストに関する情報を自オペレーティングシステムに割り当てられたオペレーティングシステム用メモリ領域に保存してこのオペレーティングシステムを一時停止状態にするサスペンド機能を有する場合に、このサスペンド機能によりこのオペレーティングシステムを一時停止状態に移行させる処理と、切り替え先のオペレーティングシステムが、サスペンド機能により一時停止状態にある自オペレーティングシステムを動作状態に移行させると共に、オペレーティングシステム用メモリ領域に保存されている自オペレーティングシステムの動作コンテキストに関する情報に基づいて自オペレーティングシステムが一時停止状態に移行する直前の動作コンテキストを回復させるレジューム機能を有する場合に、このレジューム機能を用いて当該切り替え先のオペレーティングシステムを動作状態に移行させる処理とを前記コンピュータに実行させることを特徴としている。このような構成とすれば、このプログラムをインストールした全てのコンピュータにおいて、共存させたオペレーティングシステムを、サスペンド機能及びレジューム機能を用いて高速に切り替えることが可能となる。

【0025】また、本発明は、コンピュータに、メモリ装置のメモリ領域を論理分割し、複数のオペレーティングシステムに個別に対応付けられた複数のオペレーティングシステム用メモリ領域を設ける処理と、複数のオペレーティングシステム用メモリ領域に、対応付けられたオペレーティングシステムを順次ロードする処理として、所定のオペレーティングシステムのブートロードを用いてこのオペレーティングシステムを当該オペレーティングシステムに対応付けられたこのオペレーティングシステム用メモリ領域にブートする処理と、さらにロードすべき他のオペレーティングシステムが存在する場合に、ブートした後に当該オペレーティングシステムを停止状態とする処理と、ブートしたオペレーティングシステムが停止状態となった後に、メモリ装置及びメモリ制御装置を除く他のハードウェアを初期化して他のオペレーティングシステムのロード処理へ移行する処理とを実行させるプログラムを記憶する記憶手段と、この記憶手段からこのプログラムを読み出してこのプログラムを送信する送信手段とを備えたことを特徴としている。このような構成とすれば、このプログラムをダウンロードして実行する全てのコンピュータにおいて、オペレーティングシステムを共存させることが可能となる。

【0026】ここで、プログラムにおけるブート後にオペレーティングシステムを停止状態とする処理は、このオペレーティングシステムが、自オペレーティングシステムの動作コンテキストに関する情報を自オペレーティ

ングシステムに割り当てられたオペレーティングシステム用メモリ領域に保存して当該オペレーティングシステムを一時停止状態にするサスペンド機能と、一時停止状態にある自オペレーティングシステムを動作状態に移行させると共に、オペレーティングシステム用メモリ領域に保存されている自オペレーティングシステムの動作コンテキストに関する情報に基づいて自オペレーティングシステムが一時停止状態に移行する直前の動作コンテキストを回復させるレジューム機能とを有する場合に、このサスペンド機能によりこのオペレーティングシステムを一時停止状態に移行させる処理と、このオペレーティングシステムが、サスペンド機能とレジューム機能とを有さない場合に、このオペレーティングシステムを単独でシャットダウンする処理とを含むことを特徴としている。このような構成とすれば、サスペンド機能及びレジューム機能を有さないオペレーティングシステムであっても、他のOSと共存させることができる。

【0027】さらにまた、本発明は、コンピュータに、メモリ装置のメモリ領域を論理分割して設けられた複数のオペレーティングシステム用メモリ領域に個別に対応付けられてロードされている複数のオペレーティングシステムのうちで、動作状態にあるオペレーティングシステムから他のオペレーティングシステムへの切り替え要求を受け付ける処理と、この切り替え要求を発行したオペレーティングシステムが、自オペレーティングシステムの動作コンテキストに関する情報を自オペレーティングシステムに割り当てられたオペレーティングシステム用メモリ領域に保存してこのオペレーティングシステムを一時停止状態にするサスペンド機能を有する場合に、このサスペンド機能によりこのオペレーティングシステムを一時停止状態に移行させる処理と、切り替え先のオペレーティングシステムが、サスペンド機能により一時停止状態にある自オペレーティングシステムを動作状態に移行させると共に、オペレーティングシステム用メモリ領域に保存されている自オペレーティングシステムの動作コンテキストに関する情報に基づいて自オペレーティングシステムが一時停止状態に移行する直前の動作コンテキストを回復させるレジューム機能を有する場合に、このレジューム機能を用いてこの切り替え先のオペレーティングシステムを動作状態に移行させる処理とを実行させるプログラムを記憶する記憶手段と、この記憶手段からこのプログラムを読み出してこのプログラムを送信する送信手段とを備えたことを特徴としている。このような構成とすれば、このプログラムをダウンロードして実行する全てのコンピュータにおいて、共存させたオペレーティングシステムを、サスペンド機能及びレジューム機能を用いて高速に切り替えることが可能となる。

【0028】

【発明の実施の形態】以下、添付図面に示す実施の形態

10

20

30

40

50

に基づいてこの発明を詳細に説明する。まず、本発明の概要について説明する。本発明は、メモリ領域を論理的に分割し、分割された各領域（論理メモリブロック）を複数のオペレーティングシステム（OS）に割り当てることにより、単一のコンピュータシステム内に複数のオペレーティングシステムを共存させる。

【0029】図1は、本発明により論理分割されたメモリマップの構成例を示す図であり、OS #1とOS #2の二つのOSを共存させた場合のメモリマップの構成を示す。図1を参照すると、メモリは、各OSが固定的に使用する0~1MBのメモリ領域101と、OSごとに割り当てられたメモリ領域102、103と、OSの切り替え等の作業に用いる独立したメモリ領域104とに論理分割されている。各OSは、自分に割り当てられたメモリ領域を当該コンピュータシステムにおけるメモリ領域全体と認識する。すなわち、図1に示すように、OS #1から見たメモリマップはメモリ領域101とメモリ領域102とを合わせた範囲であり、OS #2から見たメモリマップはメモリ領域101とメモリ領域103とを合わせた範囲である。なお、図示の例では、メモリ領域101を全メモリ領域の0~1MBの領域としているが、共存させるOSによっては異なるサイズや位置に設定することもできる。また、共存させるOSにおいて固定的に使用する領域が存在せず、競合するメモリ領域が生じない場合は、設けなくても良い。

【0030】また、本発明は、原則として各OSがサポートするサスペンド機能及びレジューム機能を利用してOSの切り替えを行う。すなわち、所定のOSを使用しているときには他のOSをサスペンドして一時停止の状態としておく。そして、OSを切り替える際に、それまで使用していたOSをサスペンドし、次に使用するOSをレジューム機能によりサスペンド直前の状態に復帰させる。

【0031】図20は、サスペンド及びレジューム動作を説明する図である。図20を参照すると、通常動作において、コンピュータシステムではアプリケーションプログラム、OS及びデバイスドライバが動作している。サスペンドイベントが発生し、サスペンド動作が開始すると、まず、アプリケーションプログラムの実行権が剥奪され、OSからアプリケーションプログラムに対しメッセージが同報される。また、デバイスドライバにサスペンドが通知され、デバイスドライバは活動を一時停止すると共に内部データを必要に応じて保存または破棄する。そして、ROM BIOSが呼び出される。ROM BIOSは、キャッシュフラッシュでキャッシュ内データをメモリに反映させると共に、CPUレジスタ及びマザーボードの基本デバイスのコンテキストを保存する。そして、システムメモリを低消費電力モードにして外部と電気的に隔絶する。そして、ハードウェアスタートマシンの起動を開始し、当該ハードウェアスタートマ

シンがパワーソースを自動で切断する。この状態がサスペンド状態である。

【0032】次に、レジュームイベントが発生すると、電源が切断されているデバイスに電源が投入される。通常のパワーオンと同じようにCPUリセット状態から動作開始したROM BIOSは、サスペンド状態においても電源が供給されていた所定のハードウェアのフラグでレジューム動作を確認し、システムメモリを通常モードに戻して通常アクセスを可能にする。そして、キャッシュの初期化が行われ、マザーボードの基本デバイスのコンテキストが回復または再初期化されて、CPUレジスタが回復する。次に、時刻情報を回復し、デバイスドライバに復帰を通知する。また、物理デバイスを操作するドライバは、以前のデバイスコンテキストが失われているものとして、動作を再開する。また必要ならば、当該デバイスの初期化を行う。そして、アプリケーションプログラムに復帰メッセージが同報される。サスペンドイベントによって実行権を剥奪されたアプリケーションプログラムは、自分に実行権が回復すると以前に実行権を剥奪された位置の直後から再スタートし、通常動作に戻る。

【0033】本発明では、論理分割されたメモリ領域にロードされたOSごとに上記のサスペンド及びレジューム処理が行われる。但し、通常のサスペンド状態ではCPUは稼働していないが、本発明ではCPUは継続して稼働することとなる。すなわち本発明では、所定のOSが通常のサスペンドを行ってCPUの稼働が停止する直前に、OSから独立したメモリ領域にロードされたOSの切り替えを司るモジュールに実行権を移動させるため、CPUは常に稼働状態にある。これにより、CPUの実行権は、それまで使用していたOSから次に使用するOSへ移動することとなる。つまり、ある時点では、所定のOSが通常動作にあり、他のOSはサスペンド状態となっている。そして、通常動作中のOSがサスペンド状態に移行した後、他のOSがレジューム処理により通常動作に復帰することによって使用するOSが切り替わることとなる。このように、本発明においては、各メモリ領域のOSは当該メモリ領域ごとに独立しているが、OSが切り替わる時にメモリ領域の境界を飛び越えた制御の移動が発生することになる。

【0034】図2は、本発明による複数OSの切り替えのイメージを説明する概念図である。図2を参照すると、まずROMコードがブートスタートしてOS #1をメモリ領域101、102へロードする。また、OS #2をメモリ領域101、103へロードする。このとき、メモリ領域101は二つのOS #1、#2の両方が使用するため競合する。そこで、使用しない方のOSにおけるメモリ領域101の内容は、独立のメモリ領域104へ退避させる。以後、メモリ上には、OS #1とOS #2の両方がそれぞれ割り当てられたメモリ領域に存

在し、サスペンド及びレジュームによりアクティブなOSが切り替わる。サスペンド及びレジュームによりOSの切り替えを行うことにより、高速な切り替えが可能となる。また、復帰したOSは、サスペンド直前の動作状態(コンテキスト)を復元するので、ユーザの利便性が飛躍的に向上する。なお、メモリ領域101の退避領域として、独立のメモリ領域104の替わりに二次記憶装置内に確保した退避用の記憶領域105を使用しても良い。記憶領域105は、各OSが保存されている2次記憶装置内の区画に、OS毎に用意されていても良いし

(図2ではOS毎に確保されている)、何れのOSにも属さない専用区画を二次記憶装置内に確保しても良い。専用の領域として確保することとすれば、OS毎に記憶領域105を用意する必要がない点で効率的である。

【0035】図3は、本実施の形態における複数のOSを共存させたコンピュータシステムのシステム構成を説明する図である。図3を参照すると、本実施の形態は、システム中に共存する複数のOSを統括的に制御するための統括制御部10と、OSごとに個別に用意される個別制御部20とを備える。後述するように、統括制御部10による制御は、各OSから独立して設けられたメモリ領域(例えば図1、図2のメモリ領域104)を用いて実行され、個別制御部20による制御は、対応するOSに割り当てられているメモリ領域を用いて実行される。

【0036】統括制御部10において、符号11はOS起動制御手段であり、システム起動時における各OSの起動を制御する。符号12はOS切り替え制御手段であり、ランタイムにおけるOSの切り替えを制御する。符号13は共用情報格納手段であり、OSの起動時や切り替え時に必要となる他のOSに関する情報などの各OSで共用すべき情報を包括して格納する。個別制御部20において、符号21はOS情報設定手段であり、複数のOSを共存させる上で必要な各OSに関する情報を設定する。符号22はOS切り替えインタフェース手段であり、ランタイムにおけるOSの切り替えをユーザが入力するためのユーザインタフェースを提供する。なお、OS情報設定手段21は、予め共存させるOSが固定的に決まっており、各OSに関する情報が予め統括制御部10の共用情報格納手段13に格納されているならば、必ずしも設けなくても良い。

【0037】OS起動制御手段11は、システム起動時に共存させる各OSを順次起動させる。この際、共用情報格納手段13に格納されている情報を参照して、各OSをロードすべきメモリ領域の位置やサイズを特定する。また、2番目以降のOSを起動させるためには、後述するように、OSのサスペンド機能等を用いて先に起動したOSを停止状態にした後、システムを初期化して次のOSの起動に移行する。したがって、共存させる全てのOSが起動した直後は、最後に起動したOSが動作

状態にあり、それ以前に起動した他の全てのOSが停止状態となっている。複数のOSを起動させる具体的な動作については後述する。

【0038】OS切り替え制御手段12は、ランタイムにおいて使用するOSの切り替えを行う。すなわち、動作状態にあるOSにおける個別制御部20のOS切り替えインタフェース手段22から次に使用するOSに関する情報を受け取り、動作状態にあった当該OSがサスペンド機能等により停止状態に移行した後に、当該受け取った情報に基づいて、次に使用するOSを動作状態に移行させる。停止状態にあったOSを動作状態に移行させる手段としては、OSに用意されているレジューム機能を用いたり、仮想的にブート処理を実行したりすることができる。これらの具体的な動作については後述する。

【0039】共用情報格納手段13は、共存させる各OSをロードするメモリ領域の位置及びサイズ、各OSを動作させるために最低限必要なメモリサイズ、各OSを起動するためのブートロードの二次記憶装置におけるアドレス、各OSを停止状態や動作状態に移行させるための手段、動作上必須のメモリ領域の開始物理アドレスとその長さ、さらに必要ならば重複メモリ領域を退避するための二次記憶装置内に確保された退避領域の開始アドレスといったOSの起動及び切り替えにおいて用いる情報を格納する。これらの情報は、各OSの個別制御部20のOS情報設定手段21により設定される。なお、共用情報格納手段13を含む統括制御部10は、各OSから独立したメモリ領域を用いて実行されるとしたが、共用情報格納手段13に格納される情報自体は、当該独立のメモリ領域に保持しても良いし、いずれのOSからも参照可能であることを条件にいずれかのOSのファイルとして保存しても良い。

【0040】OS情報設定手段21は、対応する各OS上で動作し、当該OSに関する情報を設定する。すなわち、上述した、共存させる各OSをロードするメモリ領域の位置及びサイズ、各OSを動作させるために最低限必要なメモリサイズ、各OSを起動するためのブートロードの二次記憶装置におけるアドレス、各OSを停止状態や動作状態に移行させるための手段、動作上必須のメモリ領域の開始物理アドレスとその長さ、さらに必要ならば重複メモリ領域を退避するための二次記憶装置内に確保された退避領域の開始アドレスといったOSの起動及び切り替えにおいて用いる情報である。また、システム起動時に、当該OSの次にいずれのOSを起動するかといった情報の設定も行う。

【0041】OS切り替えインタフェース手段22は、対応する各OS上で動作し、ユーザからの入力を受け付けて、次に使用する他のOSを指定してOS切り替え動作を開始させる。OS切り替え動作が開始した後、当該OSは、サスペンド機能などを用いて停止状態に移行する。具体的なOS切り替え動作については後述する。

【0042】図4は、本実施の形態を実現する論理分割されたメモリ装置における論理メモリブロックの構成を説明する図である。図4に示すメモリ装置は、OS # 1、OS # 2及びOS # 3の三つのOSを共存させる。したがって、OS # 1用論理メモリブロック410と、OS # 2用論理メモリブロック420と、OS # 3用論理メモリブロック430と、OSから独立した独立論理メモリブロック400とを備える。また、図示しないが、メモリ装置は、OS # 1、OS # 2及びOS # 3に共有される論理メモリブロックを備える。ただし、この共有される論理メモリブロックは、OS # 1、OS # 2及びOS # 3のうちのいずれかが動作状態にある時に、当該動作中のOS用の論理メモリブロックとして用いられる。ここでは、本実施の形態において共存させるOS # 1、OS # 2及びOS # 3は、いずれもサスペンド機能及びレジャー機能等を備えるものとする。

【0043】図4を参照すると、独立論理メモリブロック400には、統括制御部10のOS起動制御手段11に対応する機能実行部として、マルチOS初期化部401、システム仮想初期化部402、メモリ論理分割制御部403及び仮想ブート制御部404がロードされている。また、OS切り替え制御手段12に対応する機能実行部としてOS切り替え制御部405がロードされている。さらに、共用情報格納手段13に対応する機能実行部としてOS特性登録部406及びOSブートロード登録部407がロードされている。

【0044】ここで、マルチOS初期化部401は、システム起動時に、各OSを順次ブートスタートさせる。システム仮想初期化部402は、マルチOS初期化部401によるOSの起動に先だって、メモリ及びメモリ制御部を除く全てのハードウェアの初期化を行う。ハードウェアの初期化を行うのは、各OSは、自分が起動するときにはシステム全体が初期状態にあると認識しているためである。このメモリ及びメモリ制御部を除く全てのハードウェアが初期化された状態を仮想初期化状態と呼ぶ。メモリ論理分割制御部403は、独立論理メモリブロック400、OS # 1用論理メモリブロック410、OS # 2用論理メモリブロック420及びOS # 3用論理メモリブロック430の位置やサイズに関する情報が格納されている。仮想ブート制御部404は、既に1つ以上のOSがブートされた状態で、さらに他のOSをブートする際に、二次記憶装置に格納された次にブートしようとするOSのブートロードにアクセスする。そして、当該ブートロードを当該メモリに割り当てられた論理メモリブロックにロードする。OS切り替え制御部405は、ランタイムにおけるOSの切り替えを行う際に、切り替え前後の二つのOSにおいて競合するメモリ領域があれば、切り替え前のOSにおける当該メモリ領域の内容を退避させて競合が起きないようにする。そして、次に起動するべきOSを動作状態に移行させる。O

S特性登録部406は、OSが動作するための最小メモリサイズや、動作のために必須の物理メモリ領域に関する情報、OSの切り替えのために停止状態に移行する場合にサスペンド機能を用いるかどうかといったOS固有の情報を登録して保持する。OSブートロード登録部407は、各OSのブートロードの二次記憶装置上のアドレスを登録して保持する。

【0045】OS # 1用論理メモリブロック410、OS # 2用論理メモリブロック420及びOS # 3用論理メモリブロック430には、それぞれ、個別制御部20のOS情報設定手段21に対応する機能実行部として、専用セットアップユーティリティ411、421、431がロードされている。また、OS切り替えインタフェース手段22に対応する機能実行部として、OS切り替えユーザインタフェース(UI) 412、422、432がロードされている。さらに、各OSにおける停止状態と動作状態とを切り替えるのに各OSがサポートするサスペンド機能及びレジャー機能を用いることから、サスペンド制御部413、423、433及びレジャー制御部414、424、434がロードされている。さらにまた、システム全体としては動作を継続しながら、各OSが個別に単独で終了できるように、仮想シャットダウン制御部415、425、435がロードされている。なお、本実施の形態では、仮想シャットダウン制御部415、425、435は必須の構成要件ではなく、各OSを単独で終了させる必要がなければ設けなくても良い。

【0046】専用セットアップユーティリティ411、421、431は、共存させる各OSをロードするメモリ領域の位置及びサイズ、各OSを動作させるために最低限必要なメモリサイズ、各OSを起動するためのブートロードの二次記憶装置におけるアドレス、各OSを停止状態や動作状態に移行させるための手段、動作上必須のメモリ領域の開始物理アドレスとその長さ、さらに必要ならば重複メモリ領域を退避するための二次記憶装置内に確保された退避領域の開始アドレスといったOSの起動及び切り替えにおいて用いる情報を設定して、メモリ論理分割制御部403、OS特性登録部406及びOSブートロード登録部407に登録する。OS切り替えユーザインタフェース(UI) 412、422、432は、ランタイムにおけるOSの切り替えのために、切り替え先のOSを指定して切り替え動作をスタートさせる。ユーザインタフェースの種類は、グラフィカルなものやキャラクタベースのものなど、OSの形態に合わせて任意に設定することができる。さらに例えば、専用ボタンの状態変化を検出するデバイスドライバから検出された状態変化の通知を受けて、そのボタンの現在の状態に対応したOSに切り替わったり、キーボードのショートカットキーを受けてそれに対応するOSに切り替わったりするような単純なユーザインタフェースでも良い。



サスペンド制御部413、423、433及びレジューム制御部414、424、434は、OSがサポートする一般的なサスペンド機能及びレジューム機能にて実現される。仮想シャットダウン制御部415、425、435は、上述したように、動作中のシステムにおいて、自OSのみを単独で終了させる。実際の動作では、使用中のOSで終了動作を行った場合に、システム全体のシャットダウンか当該OSのみの終了かを選択し、後者を選択した場合に、仮想シャットダウン制御部415、425、435による単一OSの終了が行われる。上述したように、仮想シャットダウン制御部415、425、435は必須の構成要素ではないが、OSの切り替えにサスペンド機能及びレジューム機能を用いる代わりに、仮想シャットダウン制御部415、425、435による単一OSの終了を介して、次のOSへの切り替えを実現することができる。この場合の動作手順については後述する。

【0047】次に、本実施の形態において、OS#1、OS#2及びOS#3を共存させたコンピュータシステムを起動させる手順について説明する。図5は当該コンピュータシステムの起動の手順を説明するフローチャート、図6は当該起動時の状態遷移を説明する図である。本実施の形態では、メモリ装置上の独立論理メモリブロック400にマルチOS初期化部401、システム仮想初期化部402、メモリ論理分割制御部403、仮想ブート制御部404、OS切り替え制御部405、OS特性登録部406及びOSブートロード登録部407がロードされることにより、複数OSの共存が実現する。したがって、システム起動時の最初の一連の動作の中で、これらの機能実行部が独立論理メモリブロック400にロードされることが必要となる。これらの機能実行部をロードする手法としては種々の方法が考えられるが、ここでは、特定のOSと共にロードする方法と、最初に独立論理メモリブロック400へのロードを行う方法とを例示する。

【0048】まず、特定のOSと共に独立論理メモリブロック400へロードする方法について説明する。この手法では、システム起動時に最初にブートするOS（以下、メインOSと称す）を特定しておき、当該OSの関係ファイルとしてマルチOS初期化部401、システム仮想初期化部402、メモリ論理分割制御部403、仮想ブート制御部404、OS切り替え制御部405、OS特性登録部406及びOSブートロード登録部407のプログラムを格納しておく。そして、当該OSをブートした際に、一緒にこれらの機能実行部をロードしてしまう。この場合、2番目以降に起動するOSについては、マルチOS初期化部401、システム仮想初期化部402、メモリ論理分割制御部403、仮想ブート制御部404、OS切り替え制御部405、OS特性登録部406及びOSブートロード登録部407によりメモリ

領域の管理も含めて起動制御を行う。したがって、2番目以降に起動するOSは、自分に割り当てられているメモリ領域をOSが本実施の形態によるOSの共存の仕組みに対応している必要はない。しかし、最初に起動するメインOSだけは、独立論理メモリブロック400にマルチOS初期化部401、システム仮想初期化部402及びメモリ論理分割制御部403がロードされる前に起動するため、当該メインOSに割り当てられたメモリ領域と、自OSを起動した後にマルチOS初期化部401に制御を渡して停止状態に移行する動作手順とを、OS自身が認識していることが必要である。

【0049】図5及び図6を参照してシステム起動時の動作を説明する。本動作例では、OS#1をメインOSとして最初に起動する。システム起動時の動作の概略的な流れを示す図5のフローチャートを参照すると、まず、メインOSをブートした後（ステップ501、502）、他にブートするOSがあるか調べる（ステップ503、504）。起動するOSがある場合、次にブートするOSを指定し（ステップ505）、当該OSをロードするための論理メモリブロックを指定する（ステップ506）。そして、動作中のOSをサスペンドし（ステップ507）、次にブートするOSのブートロードを格納した二次記憶装置上の位置を特定して（ステップ508）、ブートスタートする（ステップ509）。この後、システム中に共存させる全てのOSをブートするまでステップ503からステップ509までの動作を繰り返す。システム中に共存させる全てのOSをブートしたならば、所定のOSからOS切り替え要求が発行されるか、終了命令が発効されるまで、動作中のOSによる処理が継続される（ステップ510、511、512）。

【0050】次に、図6の状態遷移図を参照して具体的に説明する。まず、システムの電源を投入してROMのブートスタートをすると、ROMコードがMBR（Master Boot Record）をロードしてメインOSであるOS#1のブートを開始する（①参照）。上記のように、メインOSであるOS#1は、自OSがロードされるべきメモリ領域を認識しているので、OS#1に割り当てられたメモリ領域（図4のOS#1用論理メモリブロック410）に自動的にロードされる。OS#1が起動すると、上述したように、メインOSであるOS#1のファイルとして格納されているマルチOS初期化部401、システム仮想初期化部402、メモリ論理分割制御部403、仮想ブート制御部404、OS切り替え制御部405、OS特性登録部406及びOSブートロード登録部407のプログラムが独立論理メモリブロック400にロードされる。したがって、これ以降のOS#2及びOS#3の起動は、マルチOS初期化部401、システム仮想初期化部402、メモリ論理分割制御部403、仮想ブート制御部404、OS切り替え制御部405、OS特性登録部406及びOSブートロード登録部40

7により制御されることとなる。

【0051】OS#1は、ブートされて動作状態となった後、他のOSを起動させるために制御をマルチOS初期化部401に渡し、サスペンド制御部413により、OS#1のサスペンド機能を用いて一時停止状態に移行する(②③参照)。マルチOS初期化部401は、まず、次のOS#2の動作環境を設定する際に、その利用領域がOS#1の利用領域と衝突する物理メモリ(図1、2の例では論理分割されたメモリ領域101)があるかどうかを、OS切り替え制御部405に問い合わせる。初期化時にOS特性登録部406からすでに共存する全てのOSにおける必須の物理メモリ領域に関する情報を取得しているOS切り替え制御部405は、そのような領域があれば、その内容(OS#1のプログラムの一部)を独立論理メモリブロック400に退避する。ここで、退避されるべきデータのサイズは共存させるOSの種類に応じて個別に定まる。したがって、独立論理メモリブロック400のメモリサイズも共存させるOSに応じて柔軟に設定される。既に述べたように、退避されるべきデータのサイズ及び競合する物理メモリの位置などの情報は、予めOS特性登録部406に登録されている。なお、以上の動作において、衝突する物理メモリの内容の退避先を独立論理メモリブロック400としたが、独立論理メモリブロック400の代わりに、図2で記述した、二次記憶装置内に設けられた退避用の記憶領域を用いても良い。上述したように、当該物理メモリの内容の退避先は、独立論理メモリブロック400とする

ことも、二次記憶装置内の記憶領域とすることもできるが、以下の説明においては、独立論理メモリブロック400に退避するものとして説明する。

【0052】次に、マルチOS初期化部401は、システム仮想初期化部402にハードウェアの初期化を実行させる。システム仮想初期化部402は、メモリ及びメモリ制御部を除く全てのハードウェアに対して、リブートでROMコードがハードウェアの初期設定をするのと同様の初期化処理を行う。これにより、OS#2がブートする際のハードウェアの初期条件を同一にする。このようにして他のOSが既にメモリ上に存在しながら次のOS#2のブートスタートが可能となった状態をシステム仮想初期化状態と呼ぶ(④参照)。

【0053】次に、マルチOS初期化部401は、OSブートロード登録部407から2番目に起動するOS#2のブートロードが格納されている二次記憶装置上のアドレスを取得する。そして、仮想ブート制御部404に制御を渡し、当該アドレスの示す区画にアクセスして当該ブートロードをメモリに展開させた後、当該ブートロードに制御を移す(⑤⑥参照)。この際、利用される主記憶メモリは、OS#2用論理メモリブロック420であるが、ブート中のOS#2は、当該メモリ領域が当該システムの持つ全てのメモリであると認識する。一般

に、OSは、起動時に当該マシンのメモリサイズやメモリアドレスが不連続ならば、そのメモリホールを認識するために全メモリマップを知る必要がある。通常、特定のROM BIOSが提供するサービスでそのメモリに関する情報を取得し、以後、その与えられた物理メモリの状況に順応してメモリを使い続ける。したがって、各OSの起動時に与えられるメモリ情報を、マルチOS初期化部401が意図的に論理分割したものに置き換えるだけで、同情報を取得した当該OSは、その論理分割メモリがマシンの持つすべての有効メモリであると認識する。

【0054】以下、同様にして、動作状態となったOS#2は、サスペンド制御部423により、OS#2のサスペンド機能を用いて一時停止状態に移行し、他のOSを起動させるために制御をマルチOS初期化部401に渡す(⑦⑧参照)。そして、競合するメモリ領域を検出してその内容(OS#2のプログラムの一部)を独立論理メモリブロック400に退避し、ハードウェアをシステム仮想初期化状態にする(⑧参照)。この後、OS#3のブートロードがOS#3用論理メモリブロック430に展開され、制御が移される(⑨参照)。これにより、3番目のOS#3が動作状態となる(10参照)。以上の動作の結果、OS#1、OS#2及びOS#3の三つのOSが全て起動し終わり、マルチOS動作環境が構築された時点で、先に起動したOS#1及びOS#2がサスペンド機能による一時停止状態となり、最後に起動したOS#3が動作状態となる。

【0055】次に、全てのOSに先立って独立論理メモリブロック400へのロードを行う方法について説明する。この手法では、独立論理メモリブロック400にロードするマルチOS初期化部401、システム仮想初期化部402、メモリ論理分割制御部403、仮想ブート制御部404、OS切り替え制御部405、OS特性登録部406及びOSブートロード登録部407を含むプログラムパッケージを独立のOSのように扱い、共存させようとする全てのOSに先立ってブートする。したがって、二次記憶装置上に専用の独立した区画が用意される。この場合、全てのOSは、上記メインOSのブート時に独立論理メモリブロック400へのロードを行う方法における2番目以降にブートするOSに相当することとなる。したがって、上記メインOSのように、割り当てられたメモリ領域と、自OSを起動した後にマルチOS初期化部401に制御を渡し、停止状態に移行する動作手順とを自ら認識しているようなOSを設ける必要はない。

【0056】システム起動時の具体的な動作については、図5において、ステップ501、502でメインOSの代わりに、マルチOS初期化部401、システム仮想初期化部402、メモリ論理分割制御部403、仮想ブート制御部404、OS切り替え制御部405、OS

特性登録部406及びOSブートロード登録部407を含むプログラムパッケージをブートする他は、メインOSを設定して起動する場合と同様である。また、図6において、OS#1をブートする前にマルチOS初期化部401、システム仮想初期化部402、メモリ論理分割制御部403、仮想ブート制御部404、OS切り替え制御部405、OS特性登録部406及びOSブートロード登録部407を含むプログラムパッケージをブートしてしまうため、ROMのブートスタートの後、独立論理メモリブロック400へのプログラムのロードが終了し、システム仮想初期化状態となる。その後、OS#1、OS#2、OS#3の順にブートを行う。各OSのブートの際の状態遷移は、上述したメインOSとしてOS#1を起動する場合と同様であるため、説明を省略する。

【0057】次に、ランタイムにおけるOSの切り替え動作について説明する。図7はOSの切り替え処理の手順を説明するフローチャート、図8は当該OS切り替え時の状態遷移を説明する図である。図5のフローチャートにおいて、ステップ511で、所定のOSのOS切り替えユーザインタフェース412、422、432からOSの切り替え要求が発行されると、図7の処理へ移行する。そして、個別制御部20において動作中のOSをサスペンドした後（ステップ701）、統括制御部10へ移って当該切り替え要求において指定されたOSを次に稼働させるOSとし（ステップ702）、当該OSに対応した論理メモリブロックを指定する（ステップ703）。そして、指定されたOSのレジューム制御部414、424、434を使って次のOSを動作状態に移行させる（ステップ704）。なお、OSの切り替えに伴って衝突する物理メモリがある場合には、その内容を独立論理メモリブロック400または二次記憶装置内の記憶領域に退避することとなるが、本動作においても、独立論理メモリブロック400に退避するものとして説明する。

【0058】次に、図4のブロック図及び図8の状態遷移図を参照して説明する。まず、OS#3が動作中である場合を考える。この状況は、前述したマルチOS動作環境の構築が完了した直後と同じ状態である。この状態で、例えば、ユーザが、OS#3のOS切り替えユーザインタフェース432によりOS#1への切り替えを要求したものとすると、OS#3で動作するサスペンド制御部433に実行権が移り、OS#3はサスペンド状態へ遷移する。続いて統括制御部10へ移って、OS切り替えユーザインタフェース432において切り替え先として指定されたOSは、独立論理メモリブロック400のOS切り替え制御部405に伝達される。サスペンド処理によりシステムのハードウェアコンテキストが保存されると、OS切り替え制御部405は、OS#1の状態をレジュームへ遷移させるためにOS#1のレ

ジューム制御部414に実行権を移す。レジューム制御部414は、ハードウェアコンテキストを、OS#1がサスペンドする際に保存した状態に回復する。サスペンドされていたOS#1のメモリコンテキストは、OS#3の動作中もそのまま保存され続けている。したがって、ハードウェアコンテキストが回復して、OS#1が以前にサスペンドに遷移したときに実行していた最後のアドレスの直後からその実行を再開すれば、OS#1にとっては単なるサスペンドからのレジュームと同様にして稼働を再開することができる。動作状態に復帰したOS#1は、レジューム処理において時刻情報を調整したり、サスペンド以前にネットワークに接続していた場合にはネットワークリンクをその時の接続状態に回復したりする。以上のように、OSをリブートすることなく、サスペンド機能及びレジューム機能を用いてOSの切り替えを行うため、きわめて高速にOSの切り替えを行うことができる。

【0059】次に、システム終了時の動作について説明する。上述したように、OS#1用論理メモリブロック410、OS#2用論理メモリブロック420及びOS#3用論理メモリブロック430のそれぞれに仮想シャットダウン制御部415、425、435がロードされている場合、動作中のOSを終了させる際に当該OSのみを終了させるのかシステム全体を終了させるのかを選択することができる。

【0060】図9は、この場合の動作手順を説明するフローチャートである。図5のフローチャートにおいて、ステップ512で、動作中のOSを終了させる要求が発行されると、図9の処理へ移行する。そして、終了の種類（当該OSのみの終了またはシステム全体の終了）をユーザに選択させるためのダイアログボックスを表示する（ステップ901）。当該OSのみを終了させる選択がなされた場合、当該OSを終了して（ステップ902、903）、他のOSを動作状態に移行させる（ステップ904）。ここで、動作状態に移行させるOSは、終了処理の中で選択させても良いし、予め設定しておくても良い。予め設定しておく場合は、固定的に特定のOSを動作状態に移行させることもできるし、OS#1を終了させたときはOS#2を、OS#2を終了させたときはOS#3を、OS#3を終了させたときはOS#1を動作状態に移行させるというように、終了させるOSとの関係で相対的に次に動作状態となるOSが定まるようにすることもできる。また、システム全体を終了させる選択がなされた場合、次にリブートが要求されればROMのブートエントリへ飛んでシステム全体を再起動する（ステップ905、906）。一方、リブートが要求されなかったならば、そのままコンピュータの電源を遮断する（ステップ905、907）。

【0061】図10は、本実施の形態を実現する論理分割されたメモリ装置における論理メモリブロックの他の

10

20

30

40

50

構成例を説明する図である。図10に示すメモリ装置は、図4と同様に、OS#1用論理メモリブロック410と、OS#2用論理メモリブロック420と、OS#3用論理メモリブロック430と、OSから独立した独立論理メモリブロック400とを備え、OS#1、OS#2及びOS#3の三つのOSを共存させる。また、図示しないが、OS#1、OS#2及びOS#3に共有される論理メモリブロックを備えることも同様である。ただし、OS#2はサスペンド機能及びレジューム機能を備えていないものとする。

【0062】図10を参照すると、独立論理メモリブロック400は、統括制御部10のOS起動制御手段11に対応する機能実行部として、マルチOS初期化部401、システム仮想初期化部402、メモリ論理分割制御部403及び仮想ブート制御部404がロードされている。また、OS切り替え制御手段12に対応する機能実行部としてOS切り替え制御部405がロードされている。さらに、共用情報格納手段13に対応する機能実行部としてOS特性登録部406及びOSブートロード登録部407がロードされている。

【0063】また、OS#1用論理メモリブロック410、OS#2用論理メモリブロック420及びOS#3用論理メモリブロック430には、それぞれ、個別制御部20のOS情報設定手段21に対応する機能実行部として、専用セットアップユーティリティ411、421、431がロードされている。また、OS切り替えインタフェース手段22に対応する機能実行部として、OS切り替えユーザインタフェース(UI)412、422、432がロードされている。さらに、OS#1用論理メモリブロック410及びOS#3用論理メモリブロック430には、OS#1及びOS#2が各OSにおける停止状態と動作状態とを切り替えるのに各OSがサポートするサスペンド機能及びレジューム機能を用いることから、サスペンド制御部413、433及びレジューム制御部414、434がロードされている。OS#2用論理メモリブロック420においては、OS#2がサスペンド機能及びレジューム機能を有さないため、サスペンド制御部及びレジューム制御部はロードされていない。

【0064】さらにまた、システム全体としては動作を継続しながら、各OSが個別に単独で終了できるように、仮想シャットダウン制御部415、425、435がロードされている。上述したように、OS#1及びOS#3においては、仮想シャットダウン制御部415、435は必須の構成要件ではない。しかし、サスペンド機能及びレジューム機能を有さないOS#2では、OSの切り替えにおいて、OS#2を単独で終了し、リブートによって動作状態に移行させるため、仮想シャットダウン制御部425をロードすることが必須である。

【0065】次に、本実施の形態において、OS#1、

OS#2及びOS#3を共存させたコンピュータシステムを起動させる手順について説明する。図11は当該コンピュータシステムの起動時の手順を説明するフローチャート、図12は当該起動時の状態遷移を説明する図である。なお、本動作例においても、メモリ装置上の独立論理メモリブロック400にマルチOS初期化部401、システム仮想初期化部402、メモリ論理分割制御部403、仮想ブート制御部404、OS切り替え制御部405、OS特性登録部406及びOSブートロード登録部407をロードする手法としては種々の方法が考えられるが、ここでは、上述したメインOSと共に独立論理メモリブロック400へのロードを行う場合の動作のみを説明する。

【0066】図11のフローチャートにおいて、ステップ1101からステップ1106までの動作は図5に示したステップ501からステップ506までの動作と同様であり、ステップ1111、1112、1113、1114及び1115の動作は図5に示したステップ508、509、510、511及び512の動作と同様である。本動作例では、OS#2がサスペンド機能及びレジューム機能を持たないため、ステップ1106でOSをロードするための論理メモリブロックを指定した後、動作中のOSをサスペンド機能にて停止状態にするのか、当該OSを終了するのかを選択する(ステップ1107)。そして、OS#1及びOS#3のようにサスペンド機能にて停止状態にする場合は、動作中のOSをサスペンドする(ステップ1108)。一方、OS#2のようにOSを終了する場合は、当該OSを単独で終了させる(ステップ1109)。続いて、何れの場合でも、システムをシステム仮想初期化状態にした後(ステップ1110)、次のブートOSのブートロードの格納位置を特定する(ステップ1111)。

【0067】次に、図12の状態遷移図を参照して具体的に説明する。本動作例では、ROMのブートスタートからメインOSであるOS#1を起動し、さらにOS#1をサスペンド機能を用いて一時停止状態に移行させた後、マルチOS初期化部401の制御の下、OS#2をブートするまでの動作(①～⑤参照)は、図6に示した動作と同様なので、詳細な説明を省略する。OS#2のブートが完了すると(⑥参照)、OS#2のOS切り替え用制御コードである仮想シャットダウン制御部425も実行状態になる。そこで、OS#2は、仮想シャットダウン制御部425を用いて単独で終了する。そして、終了処理の完了時にマルチOS初期化部401に制御を渡す(⑦⑧参照)。マルチOS初期化部401は、競合するメモリ領域の内容(OS#2のプログラムの一部)を独立論理メモリブロック400に退避し、ハードウェアをシステム仮想初期化状態にする(⑧参照)。この後、OS#3のブートロードがOS#3用論理メモリブロック430に展開され、制御が移される(⑨参照)。

これにより、3番目のOS # 3が動作状態となる（10参照）。

【0068】次に、ランタイムにおけるOSの切り替え動作について説明する。図13はOSの切り替え処理の手順を説明するフローチャート、図14は当該OS切り替え時の状態遷移を説明する図である。図11のフローチャートにおいて、ステップ1114で、所定のOSのOS切り替えユーザインタフェース412、422、432からOSの切り替え要求が発行されると、図13の処理へ移行する。まず個別制御部20において、動作中のOSをサスペンド機能により停止状態とするのか終了させるのかを判断し、当該動作中のOSをサスペンドまたは終了させる（ステップ1301、1302、1303）。次に、統括制御部10に移って、当該切り替え要求において指定されたOSを次に稼働させるOSとし（ステップ1304）、当該OSに対応した論理メモリブロックを指定する（ステップ1305）。この後、当該次に稼働させるOSがサスペンド状態にあるのか終了しているのかを調べ、サスペンドによる一時停止状態であれば、個別制御部20に移って、当該OSのレジューム機能により動作状態に移行させる（ステップ1306、1307）。一方、当該OSが終了しているならば、当該OSのブートローダを格納した二次記憶装置上の位置を特定して（ステップ1308）、システムをシステム仮想初期化状態とし（ステップ1309）、ブートスタートする（ステップ1310）。

【0069】次に、図14の状態遷移図を参照して説明する。サスペンド機能及びレジューム機能を用いるOS # 1とOS # 3との間の切り替えに関しては、図8を参照して説明した動作とほぼ同様なので説明を省略する。まず、OS # 2への切り替え動作について説明する。例えば、OS # 1の稼働中に、ユーザが、OS # 1のOS切り替えユーザインタフェース412によりOS # 2への切り替えを要求したものとすると、OS # 3で動作するサスペンド制御部433に実行権が移り、OS # 3はハードウェアコンテキストが保存され、サスペンド状態へ遷移する。続いて統括制御部10に移って、OS切り替えユーザインタフェース432において切り替え先として指定されたOSを、独立論理メモリブロック400のOS切り替え制御部405に伝達する。

【0070】OS切り替え制御部405は、まず、OS特性登録部406に登録された情報からOS # 2がシャットダウンによる切り替えを行うことを判断する。なお、本動作例では、切り替えの際にサスペンド機能及びレジューム機能を用いるOS（OS # 1、OS # 3）と、シャットダウンを用いるOS（OS # 2）とが共存しているため、次に稼働するOSがどちらを用いているかの判断を必ず行う。次に、OS切り替え制御部405は、システム仮想初期化部402に制御を渡し、メモリ及びメモリ制御部を除くシステムのハードウェアを、

ROMがOSへその実行権を引き渡す際に設定するように初期化させる。すなわち、OS # 2から見ると通常のブートであるかのような状況を作り出す。次に、OSブートローダ登録部407からOS # 2のブートローダが格納されている二次記憶装置上のアドレスを取得する。そして、仮想ブート制御部404に制御を渡し、当該アドレスの示す区画にアクセスして当該ブートローダをメモリに展開させた後、当該ブートローダに制御を移す。これにより、OS # 2のブートが開始される。

【0071】OS # 2は、ブートが開始されると、BIOS等にメモリ構成を問い合わせる。BIOSに問い合わせる場合、当該BIOSは、メモリ論理分割制御部403にフックしておく。これにより、OS # 2に割り当てられたOS # 2用論理メモリブロック420がOS # 2によって認識される。具体的には、仮想ブート制御部404が、次にブートすべきOSに対応させて、OS # 2用論理メモリブロック420を予めOS特性登録部406から読み出し、メモリ論理分割制御部403に設定しておく。OS # 2は切り替えをシャットダウンとリブートにより行うので、OS # 2が動作していないときは、結果としてOS # 2用に配分された主記憶メモリは未使用のまま残されていることになる。OS # 2のブートが完了するとOS # 1からOS # 2への切り替えが完了したことになる。

【0072】次に、OS # 2から他のOSへの切り替え動作について説明する。例えば、OS # 2の稼働中に、ユーザが、OS # 2のOS切り替えユーザインタフェース422によりOS # 3への切り替えを要求したものとすると、OS # 2は、切り替えにサスペンド機能を用いないので、OS # 2の仮想シャットダウン制御部425が、OS # 2用のOS切り替えユーザインタフェース422からの指示を受けて、シャットダウンを開始する。

【0073】OS # 2の仮想シャットダウン制御部425が、シャットダウンを完了すると、OS切り替え制御部405に制御が移る。OS切り替え制御部405は、まず、OS特性登録部406に登録された情報からOS # 3がサスペンド機能及びレジューム機能による切り替えを行うことを判断する。そこで、OS # 3のレジューム制御部434へ実行権を移して、OS # 3の状態をレジュームへ遷移させ、ハードウェアコンテキストを、OS # 3がサスペンドする際に保存した状態に回復する。サスペンドされていたOS # 3のメモリコンテキストは、OS # 2の動作中もそのまま保存され続けている。したがって、ハードウェアコンテキストが回復して、OS # 3が以前にサスペンドに遷移したときに実行していた最後のアドレスの直後からその実行を再開すれば、OS # 3にとっては単なるサスペンドからのレジュームと同様にして稼働を再開することができる。動作状態に復帰したOS # 3は、レジューム処理において時刻情報を調整したり、サスペンド以前にネットワークに接続して

いた場合にはネットワークリンクをその時の接続状態に回復したりする。

【0074】以上のように、サスペンド機能及びレジューム機能を有さないOSであっても、独立論理メモリブロック400にロードされたマルチOS初期化部401、システム仮想初期化部402、メモリ論理分割制御部403、仮想ブート制御部404、OS切り替え制御部405、OS特性登録部406及びOSブートロード登録部407による統括的な制御と、システム仮想初期化状態を介するブート処理とにより、ランタイムにおけるOSからの操作で共存するOSの切り替えを行うことができる。なお、本動作例におけるOS#2のように、サスペンド機能及びレジューム機能を有さないOSにおいてはシャットダウン及びリブートをを用いて切り替えを行うため、サスペンド機能及びレジューム機能を用いる場合に比べて切り替えに要する時間がかかる。しかしながら、常駐プログラムが少ない場合や、ブート及びシャットダウンに要する時間が短いOSであれば、サスペンド機能及びレジューム機能を用いるOSと共存させてもそれほど利便性が落ちることはない。このことに基づいて、サスペンド機能及びレジューム機能を有するOSにおいても、動作状態に復帰した際に、ハードウェアのコンテキストやメモリコンテキストを停止前の状態に戻す必要がないならば、シャットダウン及びリブートをを用いた切り替えを行うようにしても良い。

【0075】図15は、本発明により論理分割されたメモリマップの他の構成例を示す図であり、OS#1とOS#2の二つのOSを共存させた場合のメモリマップの構成を示す。図15において、各OSが固定的に使用する0~1MBのメモリ領域101とOSの切り替え等の作業に用いる独立したメモリ領域104を除くメモリ領域のうち、空白で示した領域がOS#1に割り当てられたメモリ領域102であり、斜線を付した領域がOS#2に割り当てられたメモリ領域103である。ここで、メモリ領域102とメモリ領域103との関係は、一方のOSが、自己のメモリ領域の中から一定の仮想記憶メモリを取得して、他方のOSのために保持し続けている状態である。仮にOS#2がOS#1のために仮想記憶メモリを保持する場合を考えると、メモリ領域101、104を除く領域は、初期的にはOS#2のために全て割り当てられる。そして、OS#2が当該メモリ領域の中から一定の仮想記憶メモリを取得することによって、OS#1を動作させるためのメモリ領域を確保している。

【0076】OS#1に割り当てられるメモリ領域102は仮想記憶メモリとして取得されるため、物理アドレス空間においては図15に示すようにとびとびの領域となる。これに伴って、メモリ領域103も物理アドレス空間においてとびとびの領域となっている。しかし、各OSが扱う論理アドレス空間においては、それぞれ連続

した一つのメモリ領域として扱うことができる。

【0077】図16は、本実施の形態における複数のOSを共存させたコンピュータシステムのシステム構成を説明する図である。本実施の形態においても、図3に示した実施の形態と同様に、システム中に共存する複数のOSを統括的に制御するための統括制御部10と、OSごとに個別に用意される個別制御部20とを備える。また、本実施の形態では、最初にメモリ領域を割り当てられるOSにおいて、他のOSに対しメモリ領域として仮想記憶メモリを取得して割り当てるための仮想記憶メモリ管理手段23を備える。仮想記憶メモリ管理手段23は、メモリ装置におけるメモリ領域のうち統括制御部10の機能を実現するプログラムをロードするための独立のメモリ領域(図15のメモリ領域104)を除く領域にロードされたOS上で動作し、他のOSをロードするメモリ領域として仮想記憶メモリを確保する。

【0078】図17は、本実施の形態を実現する論理分割されたメモリ装置における論理メモリブロックの構成を説明する図である。図17に示すメモリ装置は、OS#1、OS#2及びOS#3の三つのOSを共存させる。したがって、OS#1用論理メモリブロック410と、OS#2用論理メモリブロック420と、OS#3用論理メモリブロック430と、OSから独立した独立論理メモリブロック400とを備える。また、図示しないが、メモリ装置は、OS#1、OS#2及びOS#3に共有される論理メモリブロックを備える。ただし、この共有される論理メモリブロックは、OS#1、OS#2及びOS#3のうちのいずれかが動作状態にある時に、当該動作中のOS用の論理メモリブロックとして用いられる。さらに、OS#2用論理メモリブロック420とOS#3用論理メモリブロック430とは、OS#1をロードしたメモリ領域において、仮想記憶メモリとして取得されたメモリ領域にて構成される。

【0079】図17において、独立論理メモリブロック400には、統括制御部10のOS起動制御手段11に対応する機能実行部として、マルチOS初期化部401、システム仮想初期化部402、メモリ論理分割制御部403及び仮想ブート制御部404がロードされている。また、OS切り替え制御手段12に対応する機能実行部としてOS切り替え制御部405がロードされている。さらに、共用情報格納手段13に対応する機能実行部としてOS特性登録部406及びOSブートロード登録部407がロードされている。

【0080】また、OS#1用論理メモリブロック410、OS#2用論理メモリブロック420及びOS#3用論理メモリブロック430には、それぞれ、個別制御部20のOS情報設定手段21に対応する機能実行部として、専用セットアップユーティリティ411、421、431がロードされている。また、OS切り替えインタフェース手段22に対応する機能実行部として、O

S切り替えユーザインタフェース (UI) 412、422、432がロードされている。

【0081】さらに、OS #1用論理メモリブロック410及びOS #3用論理メモリブロック430には、OS #1及びOS #2が各OSにおける停止状態と動作状態とを切り替えるのに各OSがサポートするサスペンド機能及びレジューム機能を用いることから、サスペンド制御部413、433及びレジューム制御部414、434がロードされている。OS #2用論理メモリブロック420においては、OS #2がサスペンド機能及びレ  
10 ジューム機能を有さないため、サスペンド制御部及びレジューム制御部はロードされていない。

【0082】さらにまた、システム全体としては動作を継続しながら、各OSが個別に単独で終了できるように、仮想シャットダウン制御部415、425、435がロードされている。サスペンド機能及びレジューム機能を有するOS #1及びOS #3においては、仮想シャットダウン制御部415、435は必須の構成要件ではない。しかし、サスペンド機能及びレジューム機能を有  
20 さないOS #2では、OSの切り替えにおいて、OS #2を単独で終了し、リブートによって動作状態に移行させるため、仮想シャットダウン制御部425をロードすることが必須である。

【0083】また、OS #1用論理メモリブロック410には、仮想記憶メモリ管理手段23に対応する機能実行部として、仮想記憶メモリ取得及び管理部416がロードされている。仮想記憶メモリ取得及び管理部416は、OS #1上で動作し、OS #1がロードされているメモリ領域上に、他のOS (OS #2及びOS #3) をロードするための一定の領域を、仮想記憶メモリとして  
30 確保する。仮想記憶メモリの取得は、図18に示すように、OS #1のメモリマネージャに対してメモリ取得要求を発行し、当該メモリマネージャから確保したメモリ領域の先頭アドレスとそのサイズを受け取ることにより行われる。また、仮想記憶メモリ取得及び管理部416は、システム起動時にOS #1のブートに伴ってロードされる。

【0084】仮想記憶メモリは、一定の領域を固定的に常時確保しておいても良いし、他のOSへ切り替えるたびに個別に確保しても良い。また、OS #2用には常時  
40 確保しておき、OS #3用の領域はOS #3へ切り替えるときに確保するようにするというように、OSごとに別個の方法を取ることも可能である。ただし、OSを切り替えるたびに個別に確保する場合は、直接OS #1に戻ったり、他のOSに切り替えた後にOS #1に戻ったりすることによって当該OSの使用を終了した場合に、確保した仮想記憶メモリを解放する。これにより、当該OSの使用終了後のOS #1または他のOSにおいて使用できるメモリサイズが増加するため、メモリ領域を経済的に使用することができる。仮想記憶メモリの具体的

な取得方法については後述する。

【0085】本実施の形態においても、図17に示した独立論理メモリブロック400における各機能実行部をロードする手段として、特定のOSと共にロードする方法や、最初に独立論理メモリブロック400へのロードを行う方法など、種々の方法を取ることができる。ただし、本実施の形態では、OS #1は独立論理メモリブロック400を除く全てのメモリ領域を用いてロードするので、最初にOS #1をブートし、これに伴って独立論理メモリブロック400の各機能実行部をロードする手法を取ることが容易である。この場合、独立論理メモリブロック400の各機能実行部を実現するプログラムは、初期的にはOS #1のファイルとしてロードされるので、OS #1自体はロードされるメモリ領域を全く意識することなくブートすることができる。

【0086】図19は、仮想記憶メモリ取得及び管理部416による仮想記憶メモリの取得動作を説明するフローチャートである。この動作は、OS #1が停止状態に移行するためにサスペンド制御部413に制御が移った際に、当該サスペンド制御部413から呼び出される。図19を参照すると、まず取得する仮想記憶メモリのタイプをページャブル (Pagable) メモリに設定する (ステップ1901)。次に、切り替え先のOSの使用頻度、すなわち、今回使用した後しばらく使用しないのか、または度々切り替えて使用するのかを調べる (ステップ1902)。切り替え先のOSの使用頻度が高い場合は、さらに、当該切り替え先のOSがOSを切り替えるための停止状態に移行するためにサスペンド機能を用  
50 いるかどうかを調べる (ステップ1903)。そして、サスペンド機能を用いる場合は、取得する仮想記憶メモリのタイプをノン・ページャブル (Non-pagable) メモリに設定し直す (ステップ1904)。切り替え先のOSの使用頻度が低い場合や、切り替え先のOSが停止状態に移行するためにサスペンド機能を用いない場合は、取得する仮想記憶メモリのタイプはページャブルメモリのままとする。

【0087】ここで、初期的に取得する仮想記憶メモリのタイプをページャブルメモリとするのは、ページャブルメモリがOSのメモリマネージャにとって比較的確保しやすいコストの安いメモリ領域だからである。しかし、ページャブルメモリは、取得元のデバイスドライバである仮想記憶メモリ取得及び管理部416が認識しないうちに、OSのメモリマネージャによってスワップアウトされてしまう可能性がある。つまり、OS切り替え直前にページャブルメモリを確保して、そのメモリ領域を他のOSへ与えても、当該OSに戻ってきてしばらく当該OSで動作中にスワップアウトされてしまう可能性が高い。そのため、再度同様に以前のOSに切り替わる際に、以前と同じ物理メモリ領域とは限らないということである。これに対して、ノン・ページャブルメモリ

は、一旦取得すれば全く他の介入を受けない。そこで、切り替え先のOSを頻繁に使用し、かつ停止状態とする際にサスペンド機能を用いて使用していたOSコンテキストを保持したい場合は、取得する仮想記憶メモリのタイプをノン・ページャブルメモリに切り替える必要がある。さらに、たとえ1回限りの利用のための切り替えの場合でも、確保しなければならない仮想メモリ領域のサイズが大きい場合には、ページャブルメモリとして同サイズ分実際のメモリ領域が確保されない場合もある。何故なら、仮想記憶メモリ管理では一般に、実際に使用する段階で他の用途で利用中のメモリをスワップアウトして逐次確保する方式が採用されているからである。したがって、図19においてページャブルメモリの取得として記述されている例であっても、特定の場合（当該マシンの全搭載メモリサイズ、要求する仮想メモリのサイズ、その時点で動作中の他のタスクの状況など様々な要因）によっては、ノン・ページャブルメモリを要求する場合もありうる。なお、OSの使用頻度やサスペンド機能を用いるかどうかといった情報は、各OSの専用セットアップユーティリティ411、421、431において設定し、OS特性登録部406に登録しておくことができる。

【0088】取得する仮想記憶メモリのタイプが決まったならば、ロードするOSが必要とする動作メモリのサイズをメモリ論理分割制御部403から取得する（ステップ1905）。次に、取得したメモリサイズ分の仮想記憶メモリをOS#1のメモリマネージャから取得する（ステップ1906）。そして、取得した仮想記憶メモリによる切り替え先のOS用の論理メモリブロックをサスペンド制御部413に返す（ステップ1907）。

【0089】上述したように、以上の動作はOS#1が停止状態に移行する際に、サスペンド制御部413による呼び出しに応じて実行される。すなわち、OS#1のOS切り替えユーザインタフェース412からの要求で開始されるOS切り替えイベントに応じて、サスペンド制御部413によってその都度呼び出されて実行される。これは、切り替え先のOSの使用頻度が低く追加的に呼び出される（切り替えられる）だけで、当該OSのためにシステムの起動時から固定的にメモリ領域を確保しておくことが、メモリの利用効率から見て好ましくない場合に適している。これに対し、使用頻度が高く相手OSがサスペンド機能を使って切り替えを行うと判っている場合には、OS#1をブートした際に、仮想記憶メモリ取得及び管理部416の初期化ルーチンでノン・ページャブルメモリの確保を予め行っても良い。これは、予め所定のOSをOS#1と共存させることが指定されている場合、すなわち、OS#1を含む複数のOSの使用頻度にあまり差がない場合に適している。

【0090】次に、OSをロードするために確保した仮想記憶メモリを解放する処理について説明する。1回限

りで追加的に呼び出されたOSの利用が終わった場合や、当該OSがサスペンド機能を持たず切り替え時にシャットダウンとリブートを行う場合は、確保して保持しておいても非効率なので、当該OSからOS#1に戻った時点で、当該OSのために確保された仮想記憶メモリを解放する。さらに、呼び出されたOS（OS#2やOS#3）のOS切り替えユーザインタフェース422、432から当該OSの利用予定が今後ない旨の指示がユーザー自身によって切り替え指示と併せて発行された場合にも、メモリ論理分割制御部406を経てOS#1に伝達され、それを受けてOS#1のレジューム制御部414の指示により、仮想記憶メモリ取得及び管理部416がOS#1のメモリマネージャに仮想記憶メモリの解放を依頼することにより実行される。

【0091】本実施の形態におけるその他の動作、すなわち、システム起動時における各OSのブート及び独立論理メモリブロック400への機能実行部のロードと、ランタイムにおけるOSの切り替えにおけるサスペンド制御部413、433、レジューム制御部414、434、仮想シャットダウン制御部425、OS切り替え制御部405等の動作は、図5乃至図9及び図11乃至図14を参照して説明した動作と同様であるため説明を省略する。

【0092】なお、上述した各実施の形態においては、各OSへのメモリ領域の割り当てを、各OSのOS情報設定手段（専用セットアップユーティリティ）にて設定し、共用情報格納手段に登録しておいて、各OSのブート時に対応する論理メモリブロック（割り当てられたメモリ領域）の情報を参照することとした。ここで、OSと論理メモリブロックとの対応を取る手法としては種々の方法を用いることができる。例えば、予めブートの順番を決定しておき、その順番に対応させて割り振った論理メモリブロックを固定的に各OSに返すようにしても良いし、OSのID情報を設定しておき、当該IDを用いて予め登録してある論理メモリブロックを対応付けても良い。また、OSが、本発明によるOSを共存させる機能を前提として設計されているならば、そのOS自体が適当な（または最小動作）サイズのメモリ領域をシステム側に伝えて要求する方法を取っても良い。

【0093】

【発明の効果】以上説明したように、本発明によれば、単一システム内に共存させた複数のオペレーティングシステムを高速に切り替えて使用できる環境を提供することができる。

【0094】また、単一システム内に共存させた複数のオペレーティングシステムを切り替えた際に、切り替え前の状態を保存しておき、再度当該オペレーティングシステムに切り替えた際に、切り替え前の状態を復帰できるようにすることができる。

【図面の簡単な説明】



【図1】 本発明により論理分割されたメモリマップの構成例を示す図である。

【図2】 本発明による複数OSの切り替えのイメージを説明する概念図である。

【図3】 本実施の形態における複数のOSを共存させたコンピュータシステムのシステム構成を説明する図である。

【図4】 本実施の形態を実現する論理分割されたメモリ装置における論理メモリブロックの構成を説明する図である。

【図5】 コンピュータシステムの起動の手順を説明するフローチャートである。

【図6】 図5に示すコンピュータシステムの起動時の状態遷移を説明する図である。

【図7】 OSの切り替え処理の手順を説明するフローチャートである。

【図8】 図7に示すOS切り替え時の状態遷移を説明する図である。

【図9】 システム終了時の動作手順を説明するフローチャートである。

【図10】 本実施の形態を実現する論理分割されたメモリ装置における論理メモリブロックの他の構成例を説明する図である。

【図11】 本実施の形態における他のコンピュータシステムの起動時の手順を説明するフローチャートである。

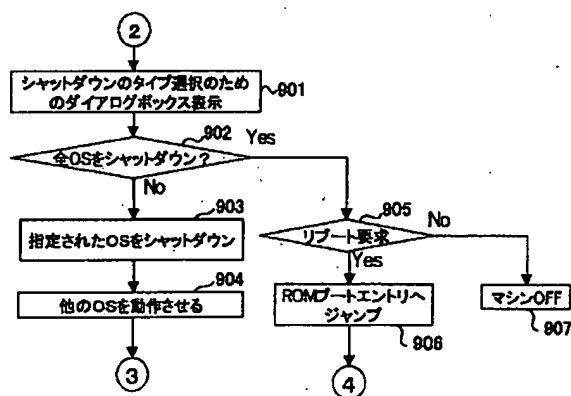
【図12】 図11に示すコンピュータシステムの起動時の状態遷移を説明する図である。

【図13】 OSの切り替え処理の手順を説明するフローチャートである。

【図14】 図13に示すOS切り替え時の状態遷移を説明する図である。

【図15】 本発明により論理分割されたメモリマップの他の構成例を示す図である。

【図9】



【図16】 本実施の形態における複数のOSを共存させたコンピュータシステムのシステム構成を説明する図である。

【図17】 本実施の形態を実現する論理分割されたメモリ装置における論理メモリブロックの構成を説明する図である。

【図18】 本実施の形態における仮想記憶メモリ取得及び管理部がOSのメモリマネージャに対してメモリ取得要求を行う様子を説明する図である。

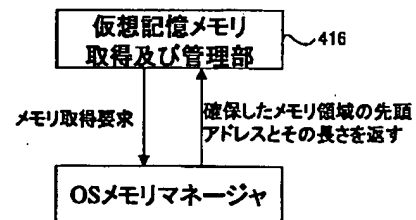
10 【図19】 仮想記憶メモリ取得及び管理部による仮想記憶メモリの取得動作を説明するフローチャートである。

【図20】 サスペンド及びレジューム動作を説明する図である。

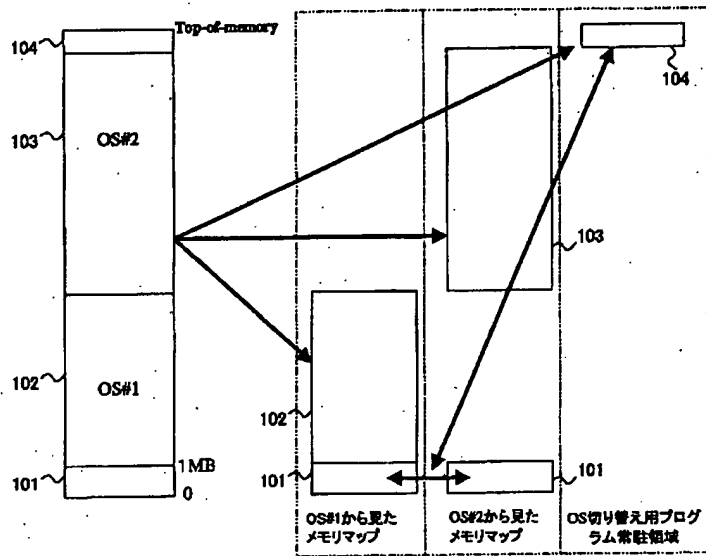
#### 【符号の説明】

10…統括制御部、11…OS起動制御手段、12…OS切り替え制御手段、13…共用情報格納手段、20…個別制御部、21…OS情報設定手段、22…OS切り替えインタフェース手段、23…仮想記憶メモリ管理手段、101、102、103、104…メモリ領域、400…独立論理メモリブロック、401…マルチOS初期化部、402…システム仮想初期化部、403…メモリ論理分割制御部、404…仮想ブート制御部、405…OS切り替え制御部、406…OS特性登録部、407…OSブートロード登録部、410…OS#1用論理メモリブロック、411、421、431…専用セットアップユーティリティ、412、422、432…OS切り替えユーザインタフェース (UI)、413、423、433…サスペンド制御部、414、424、434…レジューム制御部、415、425、435…仮想シャットダウン制御部、416…仮想記憶メモリ取得及び管理部、420…OS#2用論理メモリブロック、430…OS#3用論理メモリブロック

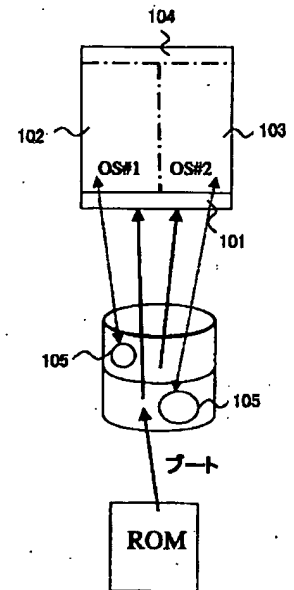
【図18】



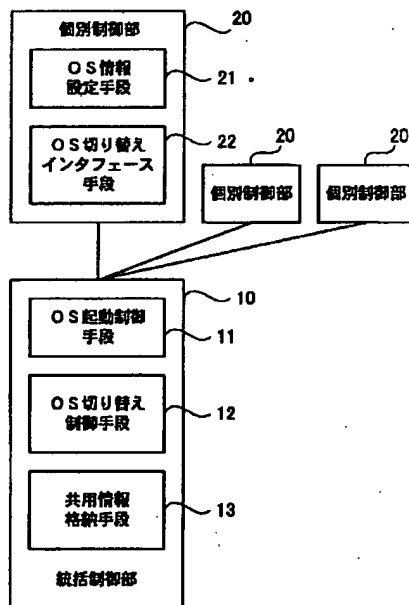
【図 1】



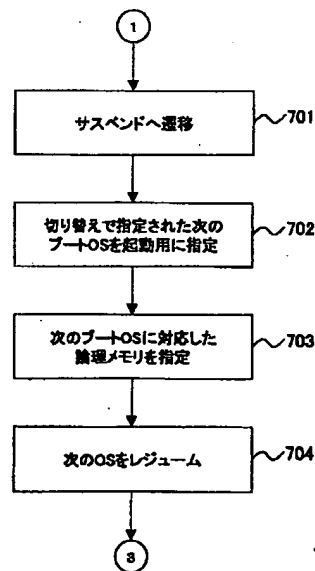
【図 2】



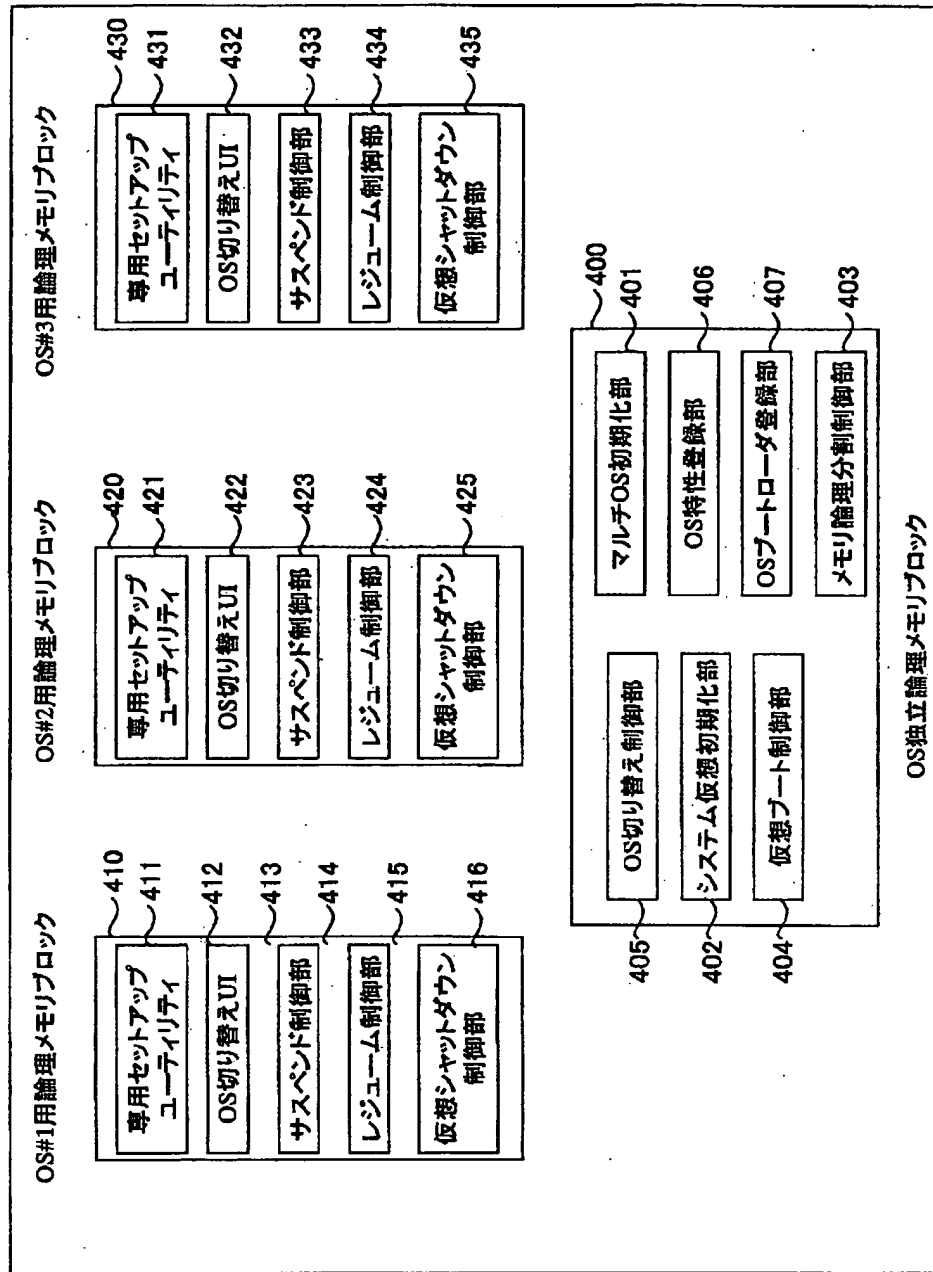
【図 3】



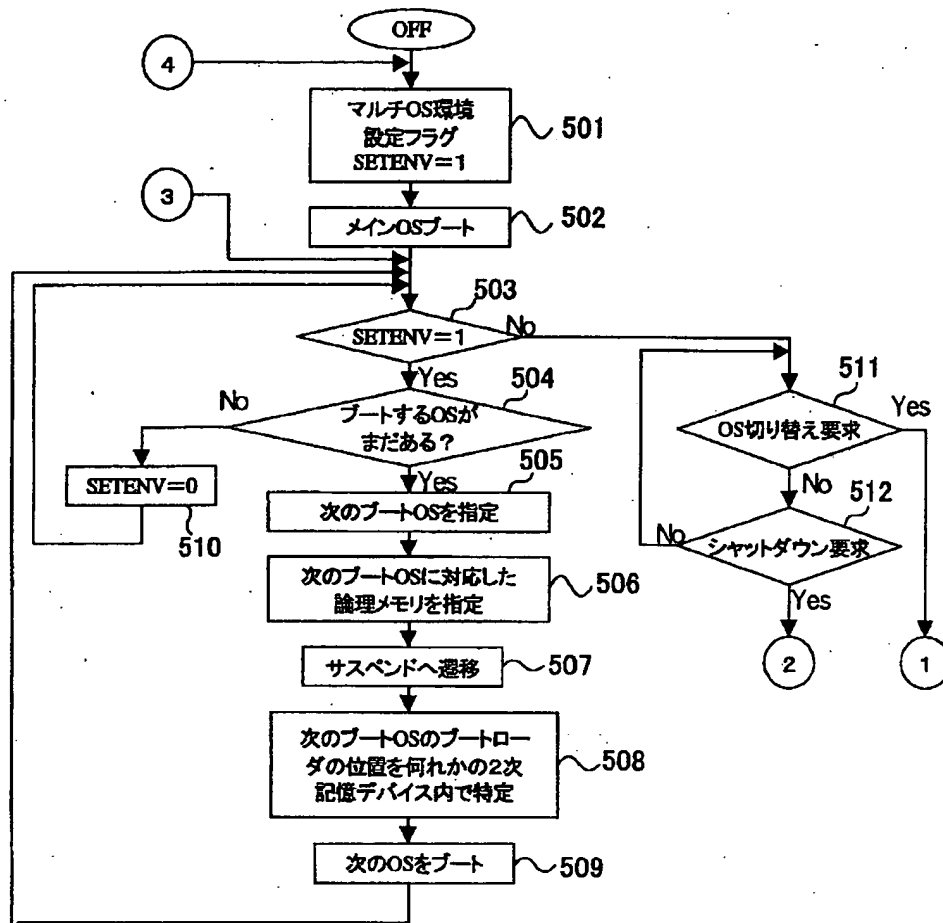
【図 7】



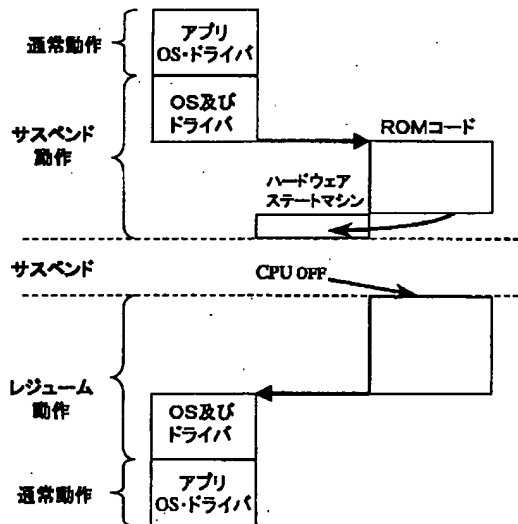
【図 4】



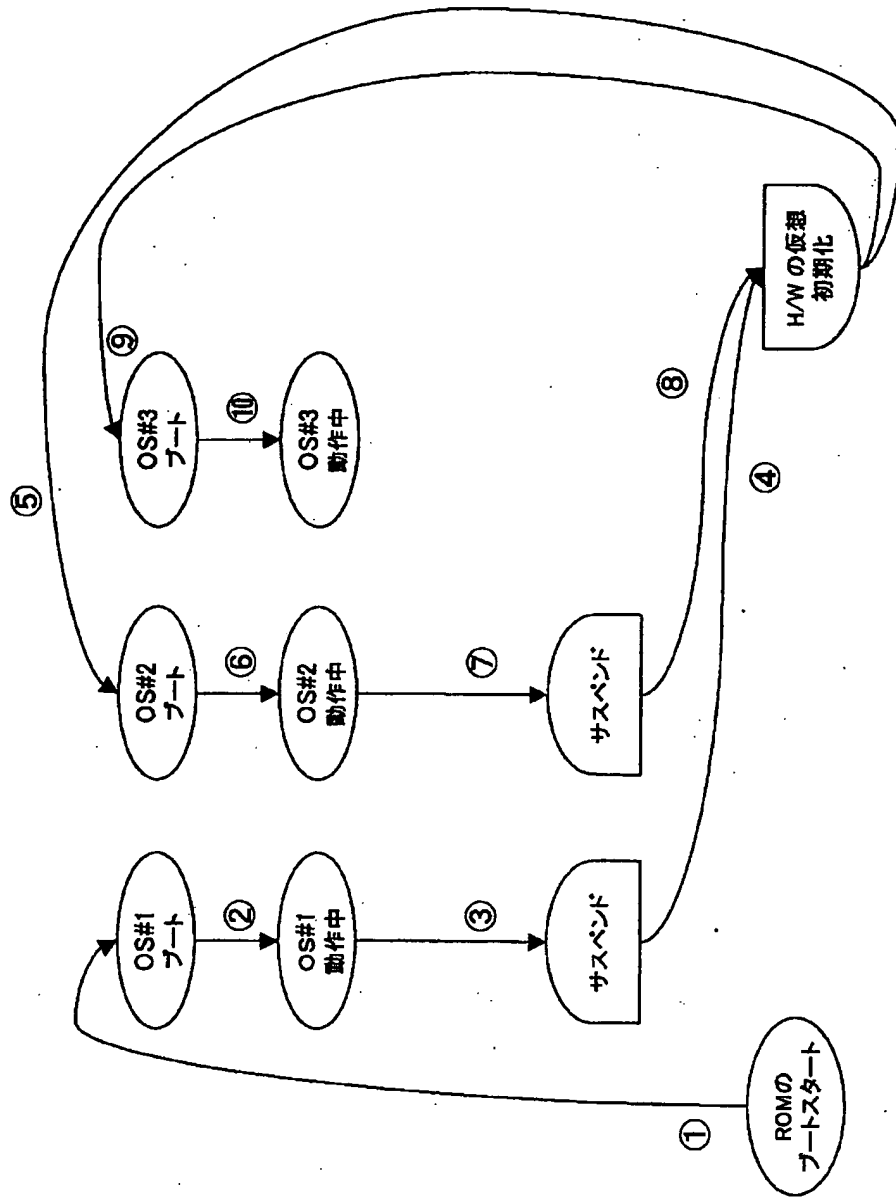
【図5】



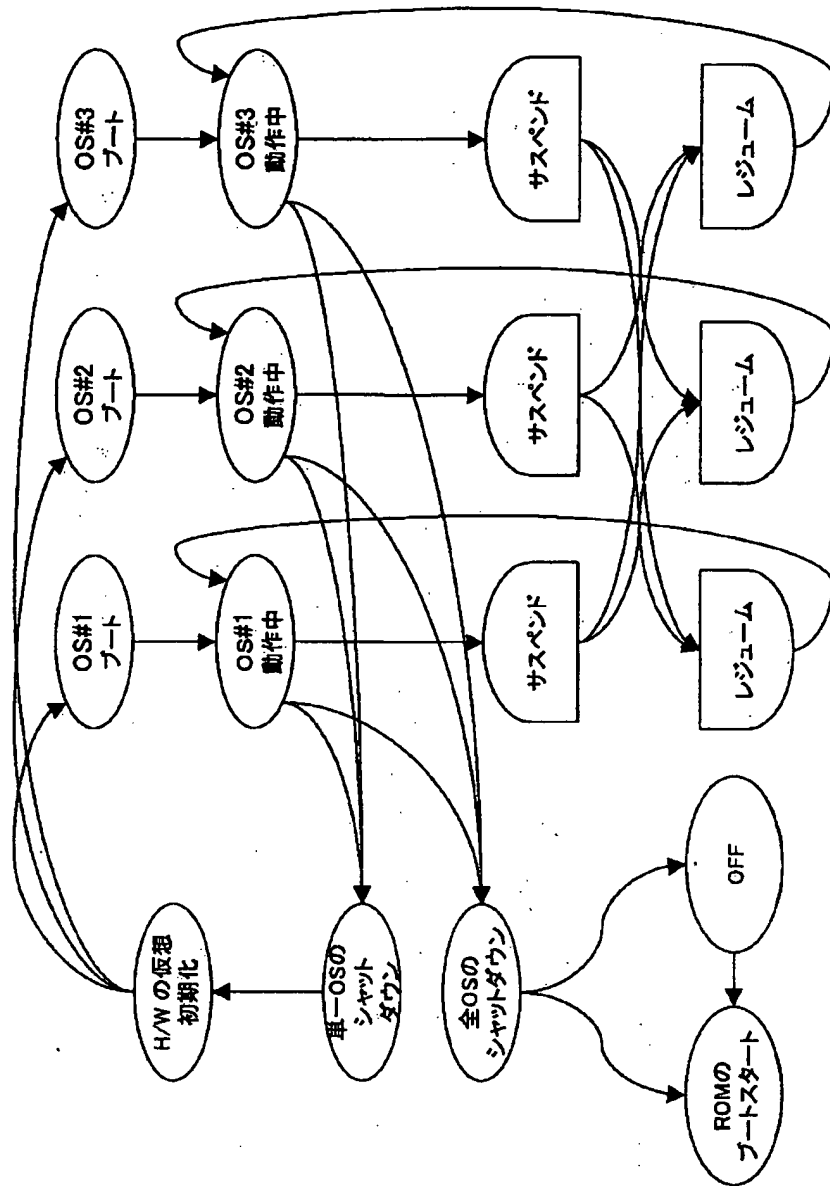
【図20】



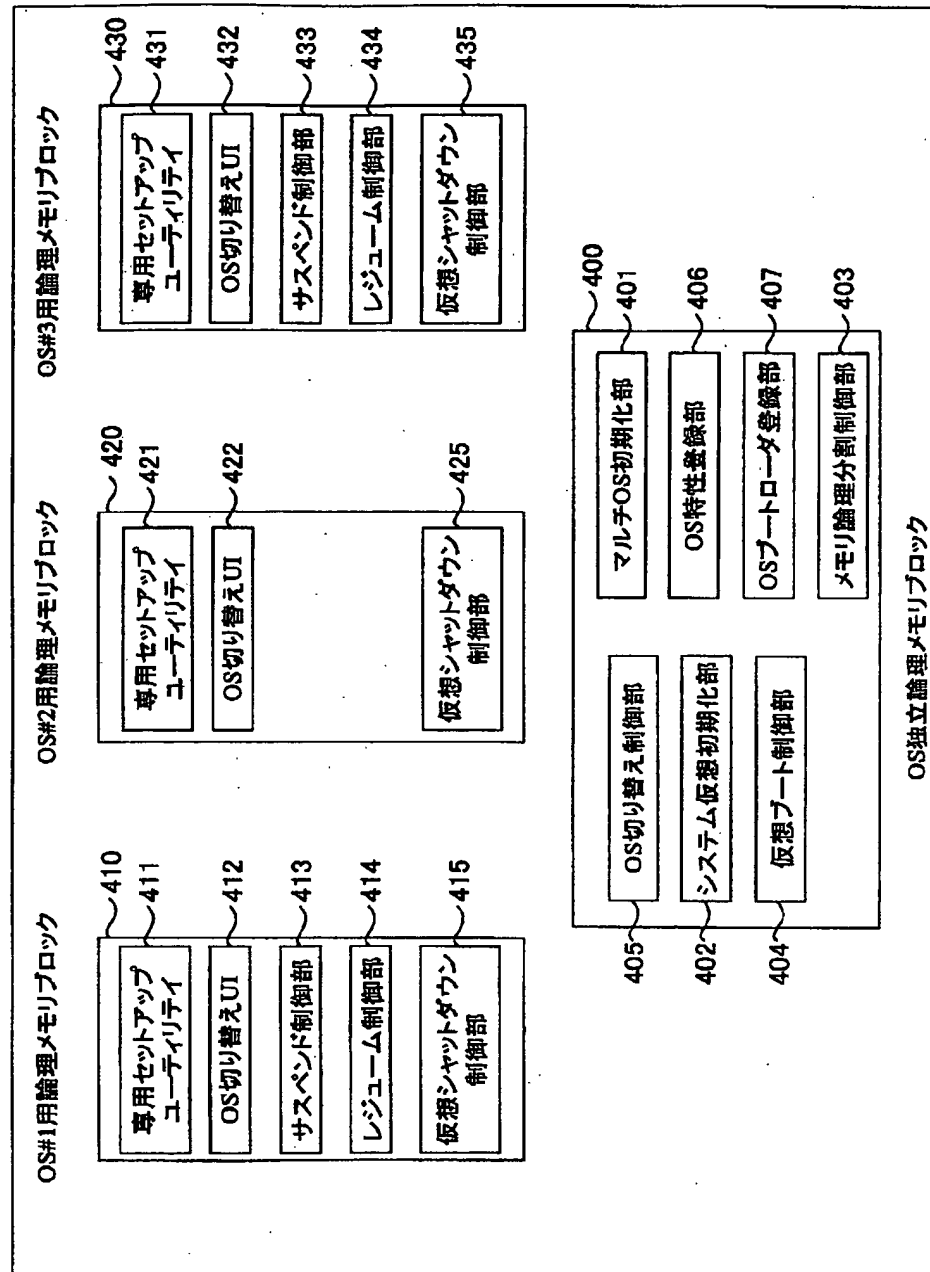
【図6】



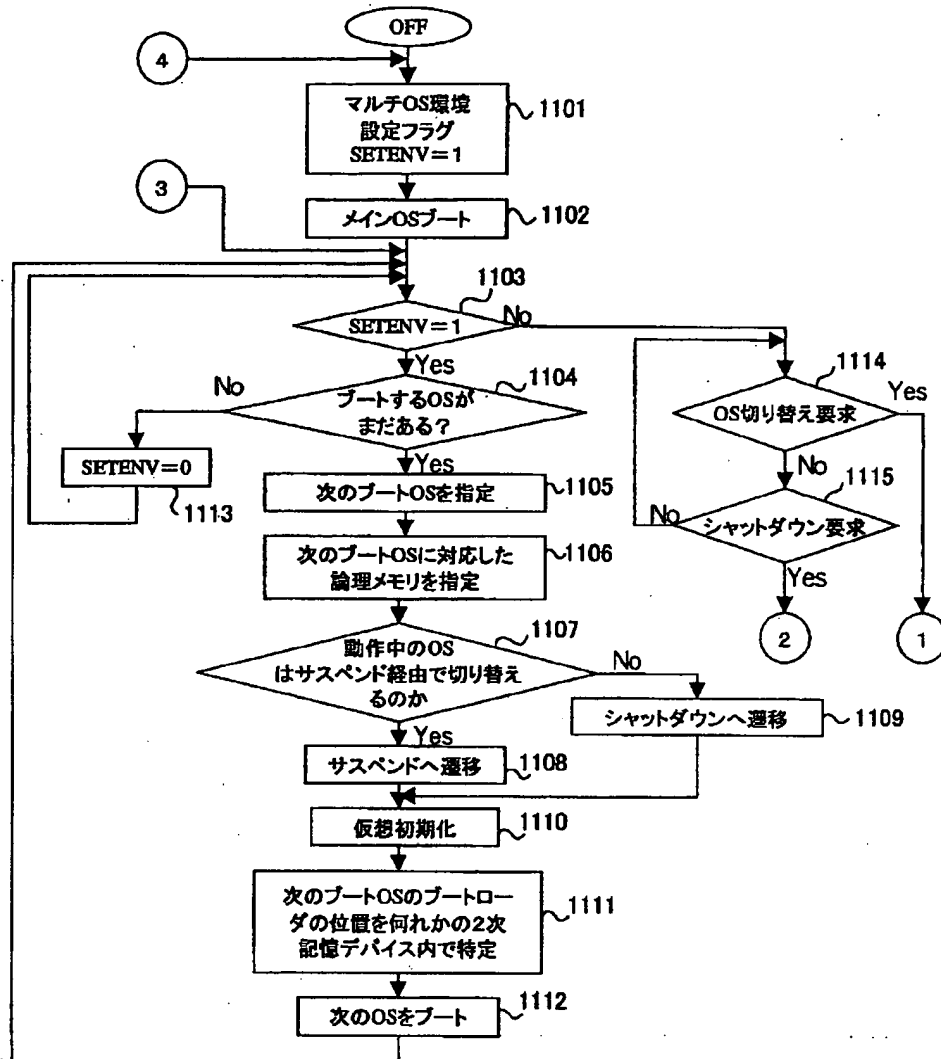
【図8】



【図10】

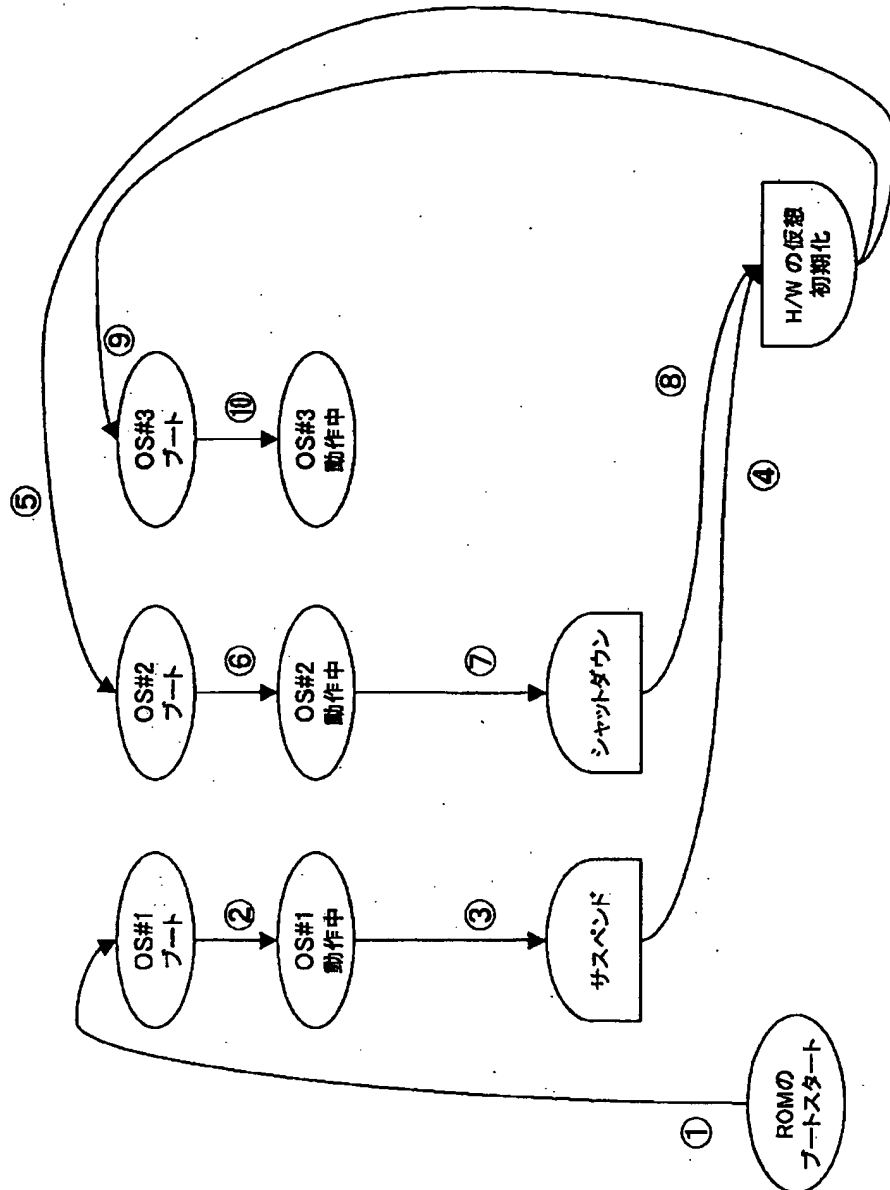


【図11】

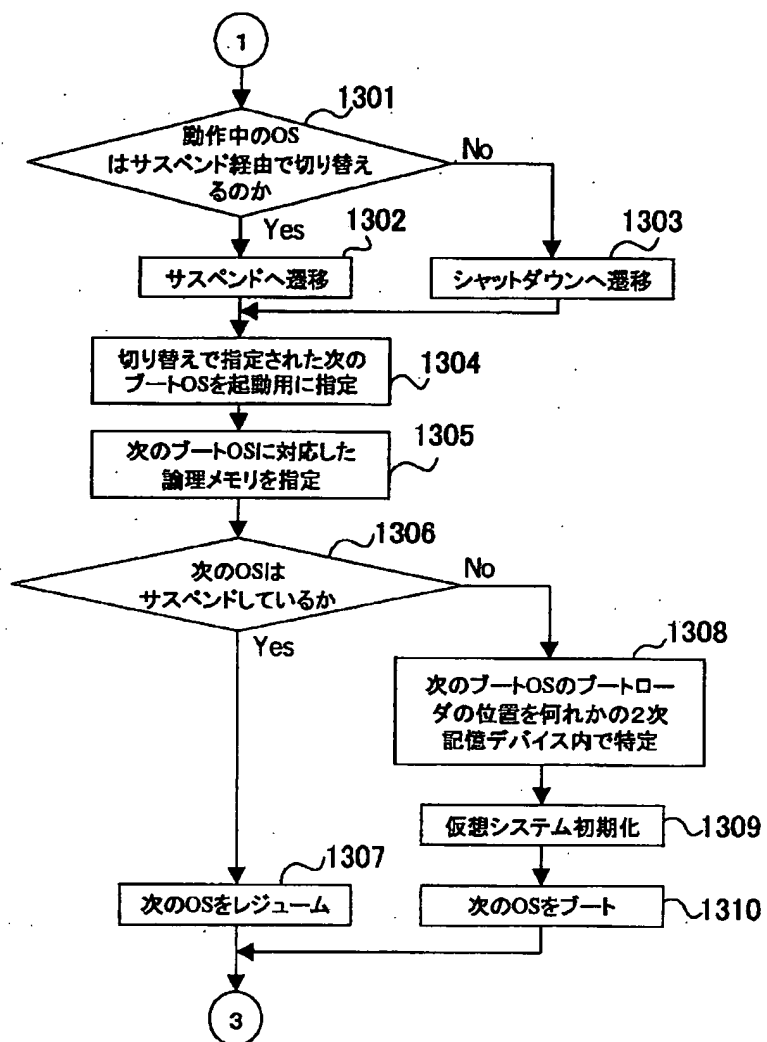




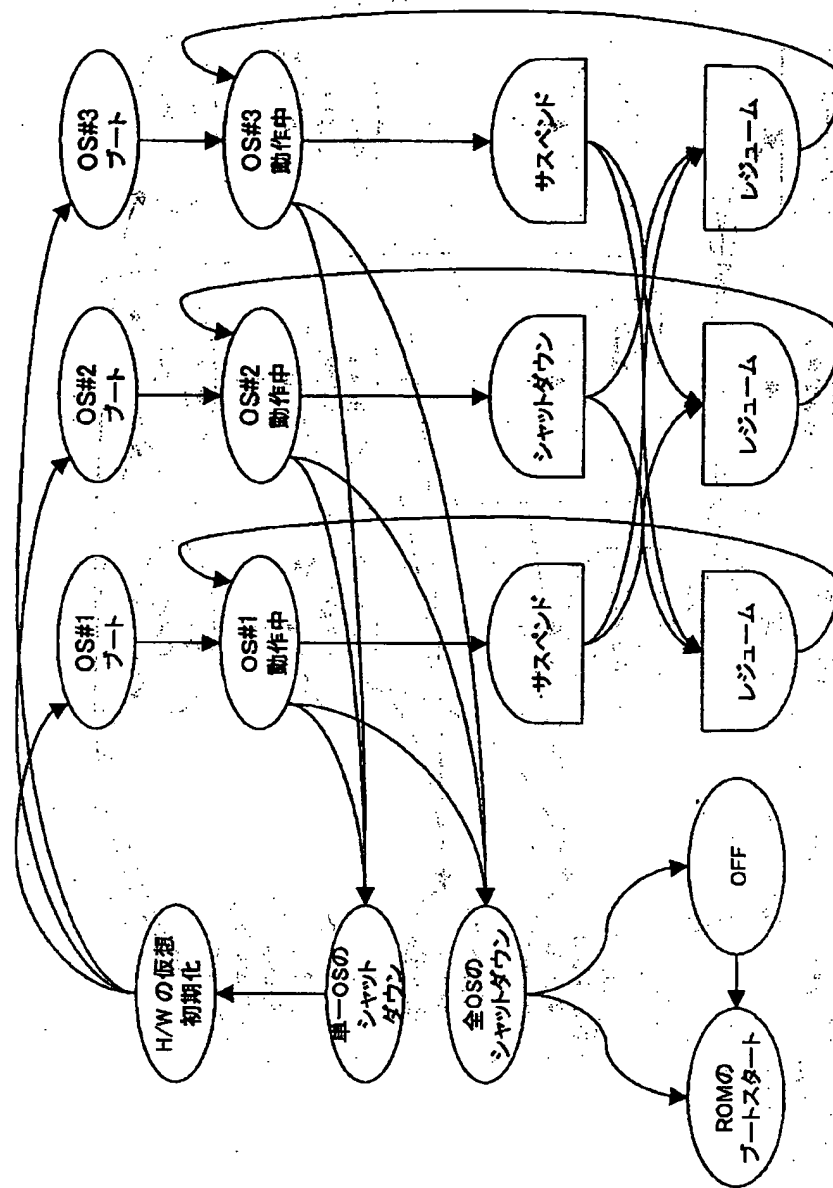
【図 12】



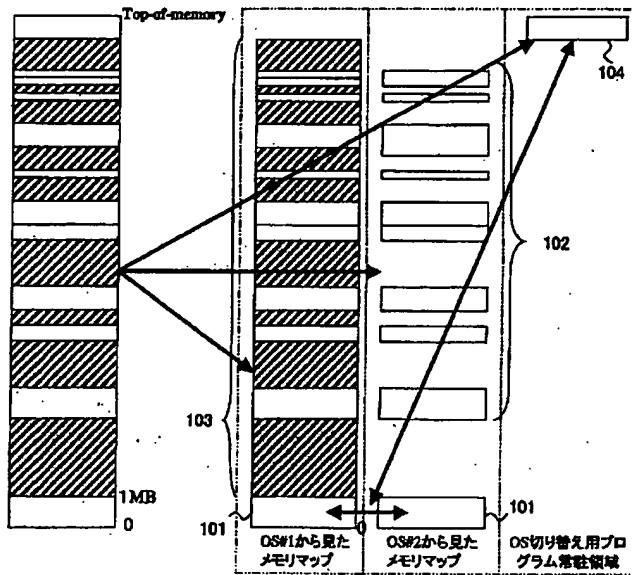
【図13】



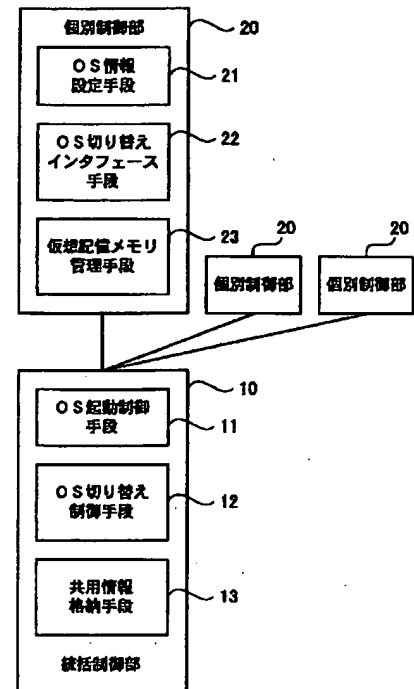
【図14】



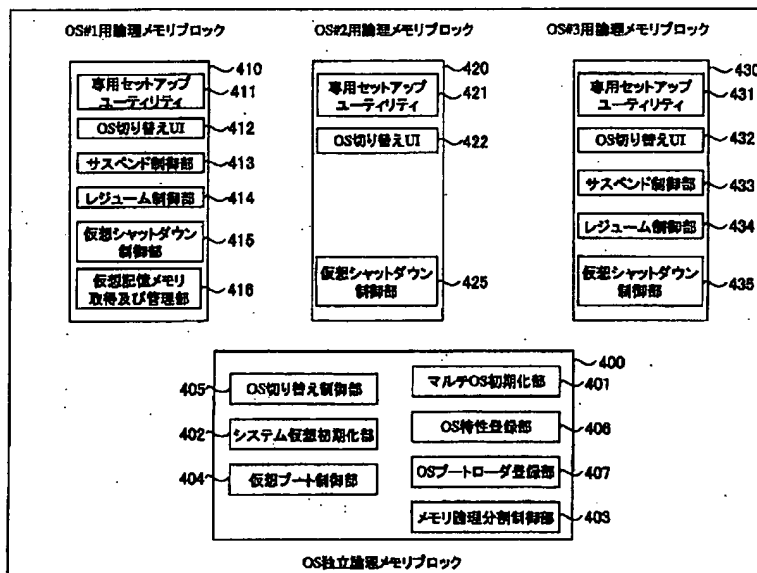
【図15】



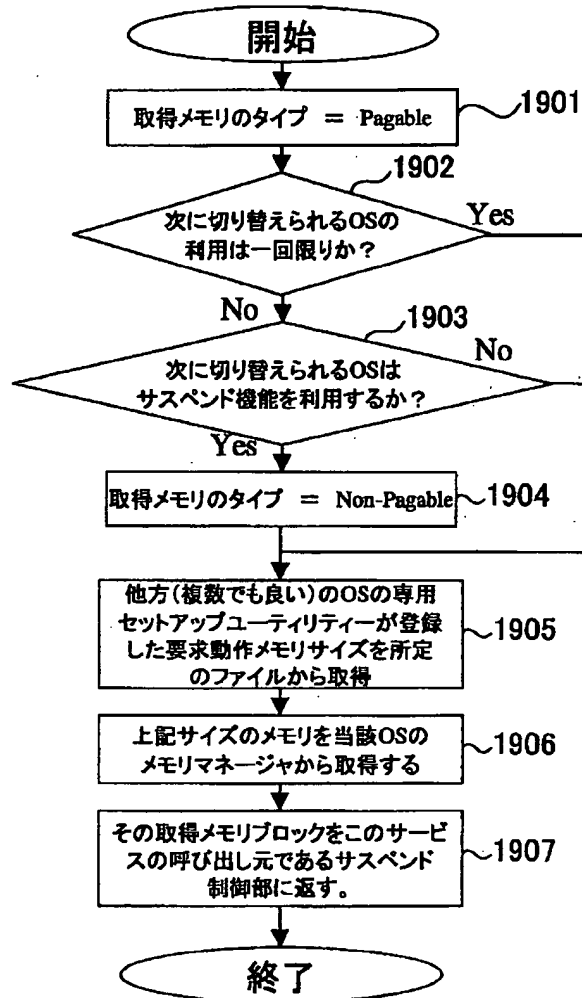
【図16】



【図17】



【図19】



フロントページの続き

(72)発明者 下遠野 享  
神奈川県大和市下鶴間1623番地14 日本ア  
イ・ビー・エム株式会社 東京基礎研究所  
内

Fターム(参考) 5B098 HH01 HH04